

Programozási versenyfeladatok tára

Nemes Tihamér
Országos Középszkolai Számítástechnikai Tanulmányi Verseny
1985–1992

Általános Iskolások számítástechnikai versenye
1991-1992

Szerkesztette: Hanák D. Péter

ITA/1143

NJSZT, Budapest, 1992



Programozási versenyfeladatok tára

Nemes Tihamér
Országos Középiskolai Számítástechnikai Tanulmányi Verseny
1985–1992

Általános Iskolások számítástechnikai versenye
1991-1992

Szerkesztette: Hanák D. Péter

NJSZT, Budapest, 1992

Az Aladdin: Alapítvány a diákinformatikáért támogatásával.

A Nemes Tihamér OKSZTV feladatsorait

dr. Ada-Winter Péter,
dr. Hanák Péter (BME) és
dr. Zsakó László (ELTE)

vezetésével az NJSZT országos versenybizottságának mindenkori tagjai dolgozták ki; közülük sok éven át

Dankó Zoltán (BME),
dr. Horváth László (ELTE),
Hunyady István (BME),
dr. Mohay Tamás (BME, IQSOFT),
Puskás Zsolt (BME),
Reé Balázs (BME),
Szabadhegyi Csaba (ELTE),
dr. Székely Jenő (ELTE),
Szlávi Péter (ELTE),
dr. Tagányi György (BME) és
Temesvári Tibor (ELTE)

vett részt a közös munkában.

Előszó

Magyarországon 1985-ben szervezte meg az első országos középiskolai programozási versenyt Nemes Tihámér Országos Középiskolai Számítástechnikai Tanulmányi Verseny (NTOKSZTV) néven a Neumann János Számítógéptudományi Társaság (NJSZT). A kétfordulós versenyen 1989-ig egy korcsoportban, 1990-től külön korcsoportban indulhatnak a 14-15. ill. a 16-19 éves tanulók (azaz a középiskolák I.-II., ill. III.-V. osztályos diákjai). A verseny első tíz helyezettje – a többi országos középiskolai tanulmányi versenyhez hasonlóan – érettségi, ill. felvételi kevezményben részesül, és közülük válogatjuk ki az 1989 óta ugyancsak évente megrendezett Nemzetközi Informatikai Diákolimpia magyar résztvevőit is. A Nemes Tihámér verseny, szerénytelenség nélkül megállapíthatjuk, népszerűvé vált a diákok körében: az utóbbi években 260-300 magyarországi középiskola kb. 2000-2000 I.-II., ill. III.-V. osztályos diákja vett részt a verseny első, iskolai fordulójában. Közülük kb. 40 fiatalabb és 60 idősebb korcsoportú diák jutott tovább a második fordulóra. Említést érdemel, hogy a környező országokban élő, magyar anyanyelvű vagy magyarul jól beszélő diákok közül is egyre többen érdeklődnek a verseny iránt, ill. vesznek részt a versenyen; az Erdélyi Magyar Műszaki Tudományos Társaság szervezésében tavaly már kb. 500 diák indult, és közülük került ki az I.-II. osztályosok korcsoportjának győztese.

Az NJSZT országos versenybizottsága sokak kívánságát teljesíti most azzal, hogy önálló kötetben megjelenteti az eddigi versenyfeladatokat. Reméljük, hogy ezzel is segíteni tudjuk a leendő versenyzőket a felkészülésben, a tanáraikat pedig a számítástechnikai foglalkozások és szakkörök megtartásában. Tudjuk, hogy ezzel csak az első lépést tesszük meg, hiszen a feladatsorok megoldásának és értékelésének részletesebb magyarázata, a második forduló programozási feladatok analízise, a megoldási lehetőségek megvitatása és nem utolsósorban kész programok bemutatása nagyon hasznosak lennének, de ennél többre jelenleg nem vállalkozhattunk. Adósak vagyunk a Nemzetközi Informatikai Diákolimpiák versenyfeladatainak közreadásával is.

Ebben a kötetben az 1985-1992 közötti Nemes Tihámér versenyek, ill. – az ELTE-n dolgozó kollégáink jóvoltából – az 1991-es és 1992-es országos általános iskolai számítástechnikai versenyek programozási feladatait ismertetjük. A kötet második részében a dolgozatokat javító tanároknak szóló megoldási és értékelési útmutatókat is közöljük.

1988-tól kezdve az alábbi kéréssel fordultunk a javító tanárokhöz:

Kérjük a tisztelt tanár kollégákat, hogy a dolgozatokat az egységes értékelés érdekében szigorúan az alábbi útmutató szerint pontozzák, a megadott részpontoszámokat ne bontsák tovább! Vagyis ha egy részmegoldásra pl. 3 pontot javasolunk, akkor arra vagy 0, vagy 3 pont adható. (Természetesen az útmutatótól eltérő megoldások is lehetnek jók.) Ha különösen értékesnek tartanak egy (részmegoldást, akkor arra inkább jól felismerhetően pluszpontot adjanak, és azt az összpontszámban is külön tüntessék föl!

A versenyekre készülő diákoknak természetesen azt javasoljuk, hogy mielőtt a közölt megoldásokat és értékelést elolvassák, saját maguk, önállóan próbálják megoldani a kitűzött feladatokat. Nem törekedtünk szövegűhűségre: ahol az eredeti megfogalmazást pontatlannak vagy hibásnak találtuk, módosítottunk a szövegben. Reméljük, hogy ezzel sikerült csökkenteni a hibás feladatok számát, de nem garantálhatjuk, hogy már nincsenek rossz vagy félreérthető megfogalmazások és megold-

dások. Nem utolsó sorban a kötet megjelentetését hátráltatta volna, ha újból és újból ellenőrizzük a szöveget. Ezért is hálásan fogadunk minden olyan kritikai észrevétel és kiegészítést, ami a feladatgyűjtemény esetleges későbbi kiadásait jobbá teszi. Címünk:

"Programozási versenyfeladatok"
NJSZT országos versenybizottsága
1054 Budapest, V. Báthori u. 16.
Levélcíme: 1368 Budapest 5, Pf. 240
Telefax: (1) 131-8140
E-levele: nttv@inf.brme.hu

A függelékben a Nemes Tihamér verseny szervezésével és ütemezésével, valamint a versenyzőktől 1993-ban elvárt ismeretekkel kapcsolatos tudnivalókat foglaljuk össze.

A Nemes Tihamér verseny 1985-1990 közötti feladatsorai néhány korábbi, ma már nehezen beszerezhető kiadványban is megtalálhatók:

1. Számítástechnikai feladatok 2000-ig I-II. OMIKK, Bpest, 1988. (Szerk. dr. Hetényi Pálné.) Második kötetében az 1985-88 közötti feladatsorok találhatóak meg.
2. Számítástechnika versenyfeladatok I. Baranya megyei Pedagógiai Intézet, Pécs, 1988. (Szerk. Dr. Poronyi Gábor) A kötet ugyancsak az 1985-88 közötti feladatsorokat tartalmazza.
3. Számítástechnika versenyfeladatok III. Baranya megyei Pedagógiai Intézet, Pécs, 1990. (Szerk. Dr. Poronyi Gábor) A kötet az 1989-es és 1990-es feladatsorokat tartalmazza.

Nyilvánvaló, hogy a versenyfeladatok összeállítása és e feladatgyűjtemény közreadása igen sok ember több éves, önzetlen munkájának az eredménye. Az alábbi névsor nem teljes, csak azokat a kollégákat, továbbá volt és jelenleg egyetemi hallgatókat soroljuk fel, akik hosszú időn át vettek részt az országos versenybizottság munkájában: dr. Ada-Winter Péter, Dankó Zoltán (BME), dr. Horváth László (ELTE), Hunyady István (BME), dr. Mohay Tamás (BME, IQSOFT), Puskás Zsolt (BME), Reé Balázs (BME), Szabadhegyi Csaba (ELTE), dr. Székely Jenő (ELTE), Szlávi Péter (ELTE), dr. Tagány György (BME), Temesvári Tibor (ELTE), dr. Zsakó László (ELTE). Veisz Imre (BME); feladatok gépelésében volt segítségemre. De nemcsak nekik tartozom köszönettel hanem elsősorban feleségemnek, Anikónak és fiamnak, Dávidnak, hogy elviselték és elviselik a versenyek szervezéséből, a feladatsorok összeállításából és most a közreadásukból közvetve rájuk is háruló terheket.

Budapest, 1992. november 30.

Hanák D. Péter

1985. Első fordulóElső-ötödik osztályosok

A programok, programrészletek HT-1080Z számítógépre készültek.
Összesen: 60 pont.

1. feladat (4 pont)

Mi a funkciója az alábbi szubrutinnak, ha M1, M2 és M3 átadott, ill. átvett paraméterek?

```
500 IF M1 < M2 OR M1 < M3 THEN IF M2 > M3 AND M2 > M1 THEN
      S = M1 : M1 = M2 : M2 = S ELSE S = M1 : M1 = M3 : M3 = S
510 IF M2 < M3 THEN S = M2 : M2 = M3 : M3 = S
520 RETURN
```

2. feladat (5 pont)

Mit csinál a következő program?

```
10 INPUT N: IF N < 2 OR N <> INT(N) THEN 10
20 FOR I = 2 TO N
30 IF N/I <> INT(N/I) THEN 70
40 PRINT I
50 N = N/I
60 IF N/I = INT(N/I) THEN 50
70 NEXT I
80 STOP
```

3. feladat (6 pont)

A következő programrészlet két növekvően rendezett számsorozat közös elemeit írja ki. Mi a feltétele annak, hogy egy elemet se írjon ki többször?

```
100 I = 1 : J = 1
110 IF I > N OR J > M THEN 150
120 IF A(I) = B(J) THEN PRINT A(I)
130 IF A(I) < B(J) THEN I = I + 1 ELSE J = J + 1
140 GOTO 110
150 _
```


4. feladat (6 pont)

Az alábbi két program mit ír ki, mikor áll meg, és van-e különbség a kapott eredményekben? A válaszokat indokold!

1. program

```
20 K% = 0 : L% = 0
30 K% = K% + L%
40 L% = 2 * L%
50 PRINT K%, -K% - 1
60 GOTO 30
```

2. program

```
110 K% = 1 : L% = 1 : J% = -(K% + L%)
120 PRINT K%, J%
130 L% = 2 * L%
140 K% = K% + L%
150 J% = -(K% + 1)
160 GOTO 120
```

5. feladat (6 pont)

Milyen hibák vannak a következő rendező szubrutinban?

```
100 DIM A(N)
...
200 REM RENDEZÉS
210 FOR I = N TO 1 STEP -1
220 FOR J = 1 TO I
230 IF A(J) > A(J+1) THEN X = A(I) : A(I+1) = A(I) : A(I) = A(I+1)
240 NEXT J
250 NEXT I
260 _
```

6. feladat (8 pont)

A következő szubrutin szövegek rendezett sorozatába beilleszt egy új szöveget; helyesen működik. Milyen kezdőértékadást kell alkalmazni ahhoz, hogy a szubrutin akkor is helyesen működjék, amikor a rendezett sorozatban még egyetlen elem sincs?

```
1000 REM BEILLESZTÉS (X$)
1010 N = N + 1 : T$(N) = X$
1020 B = 0 : A = T(B)
1030 IF A = -1 THEN 1070
1040 IF X$ <= T$(A) THEN 1070
1050 B = A : A = T(B)
1060 GOTO 1030
1070 T(B) = N : T(N) = A
1080 RETURN
```


7. feladat (10 pont)

Mit csinál a következő szubrutin? Mi a helyes működés feltétele? Indokold meg a 60-as és a 70-es sort! Miért szerepel a 110-es sorban H+1?

```

50 H1 = X2 - X1 : H2 = Y2 - Y1
60 I1 = ABS(H1) : I2 = ABS(H2)
70 IF I1 > I2 THEN H = I1 ELSE H = I2
80 X = X1 : Y = Y1
90 IF H = 0 THEN SET(X,Y) : RETURN
100 S1 = H1 / H : S2 = H2 / H
110 FOR I = 1 TO H + 1
120 SET(X,Y)
130 X = X + S1 : Y = Y + S2
140 NEXT I
150 RETURN

```

8 feladat (15 pont)

Zoli mérnök barátjától 16 nyomógombos, ún. hexadecimális billentyűzetet kapott ajándékba. "A billentyűk állapotát két függvényhívással, INP(60%) -kal és INP(61%) -kal kérdeztetheted le" – magyarázta a mérnök. – "Talán tudod, INP inputot, bevittelt jelent, 60 és 61 pedig perifériacímek. Ha a lekérdezés pillanatában bármelyik billentyű le van nyomva, a neki megfelelő bit 1 lesz a függvény előállította szavas érték alsó byte-jában; a felső byte mindenképpen 0 lesz:

függvényhívás		függvényérték							
1.	0. bitje	7	6	5	4	3	2		
INP(60%)		"7"	"6"	"5"	"4"	"3"	"2"	"1"	"0"
INP(61%)		"F"	"E"	"D"	"C"	"B"	"A"	"9"	"8"

Zoli olyan szubrutint írt, amely bármelyik billentyű leütésére a neki megfelelő decimális számot állítja elő az A% változóban:

```

1000 B% = INP(61%) * 256% + INP(60%)
1010 IF B% = 0% THEN 1000
1020 A% = -1%
1030 IF B% <> 0% THEN B% = B% / 2% : A% = A% + 1 : GOTO 1030
1090 RETURN

```

Később Zoli a szubrutint úgy akarta módosítani, hogy az a decimális érték helyett a megfelelő karakter (0, ..., 9, A, ..., F) ASCII-kódját tegye A%-ba. E célból a rutinba a következő sort szúrta be:

```
1040 A% = A% + ASC("0")
```

és kicsit csodálkozott, mert nem egészen az történt, amit várt. Amikor pedig a szubrutint alaposabban kipróbálta, észrevette, hogy az "furcsán" működik, ha az "F" billentyűt és egy vagy több másikat egyszerre üt le. Átgondolva a dolgot, az 1020-as és 1030-as sort így módosította:

```

1020 A% = 15%
1030 IF B% > 0% THEN B% = B% * 2% : A% = A% - 1 : GOTO 1030

```

Válaszolj a következő kérdésekre!

1. Mit csinál a szubrutin, amíg nem ütnek le billentyűt?
2. Mit csinál a szubrutin az 1040-es sor hatására?

- Javítsd ki úgy a fenti szubrutint, hogy az valóban a leütött karakter ASCII-kódját te-gye az A% változóba!
- Mit tesz a szubrutin első változata az A% változóba, ha egyszerre két vagy több billentyűt ütnek le?
- Milyen furcsaságot tapasztalt Zoli, amikor az "F" billentyűvel egyszerre ütött le egy vagy több másikat? Mi a jelenség magyarázata?
- Mutasd meg, hogy az új 1020-as és 1030-as sorok hatására a szubrutin minden esetben jól fog működni!

1985. Második forduló

Első-ötödik osztályosok: Darazsak

A gázok kiterjedését a gimnáziumi I. oszt. fizika tankönyv az alábbi módon modellezi: "Rajzold le két egymásba nyíló, teljesen egyforma szoba alaprajzát! Vágj ki papírból 6 darazsat, számozd meg őket 1-től 6-ig, és tedd mindet az egyik szobába! 'Röpködjenek' a darazsak teljesen rendszertelenül az egyik szobából a másikba! A véletlenszerű átröppenést dobókocka segítségével modellezd! Amelyik darázs számát dobod, az repül át a másik szobába. Jegyezd fel minden dobás után, hány darázs van a szobákban! Legalább 30 dobás után számold meg, hányszor adódott a (6,0), hányszor az (5,1), ..., hányszor a (0,6) állapot!"

Készíts a feladat számítógépes megoldására olyan programot, amely legfeljebb 16 darázs esetén szemlélteti a darazsak röpködését és eloszlását!

Jelölések:

- D - Darazsak száma,
- D(l) - az l-edik darázs melyik szobában van (1 vagy 2) (l = 1, ..., D),
- T - az aktuális idő (T = 1, 2, ...),
- X - hány darázs van adott T időpillanatban az első szobában,
- S(X) - hányszor volt X db darázs az első szobában (X = 0, ..., D).

A feladatot az alábbi, jól elkülönített (és önállóan értékelhető) részekből építsd fel:

- Szimulációs lépés (a fenti változók és tömbök kezelése).
- A darazsak megjelenítése a képernyőn:
 - a képernyőn látható legyen T és X értéke,
 - a képernyőn jelenjen meg a két szoba, és bennük a darazsak (pl. egy-egy ponttal jelképezve).
- A program felhasználójával való kapcsolat:
 - megadható legyen a darazsak száma (D) és kezdetben az első szobában lévő darazsak száma,
 - a felhasználó menet közben bármikor (!) avatkozhat be, és ilyenkor a következőket tehesse:
 - befejezheti a program használatát,
 - összegző oszlopdiagramot (hisztogramot) kérhet (l. 5. részfeladat),
 - üzemmódot változtathat:
 - a) a program minden röptetés után egy billentyű lenyomására várjon,
 - b) folyamatos legyen a röpködés,
 - c) a program megjelenítés nélkül, "gyorsan" végezze a szimulációs lépéseket.

4. Grafikon készítése a darazsak számáról az első szobában:

- a grafikonon mindig az utolsó 50 időegység állapota látszódjon (ennek a leg-egyszerűbb megoldása az, ha az 50. oszlop után újra az elsőbe kezd rajzolni a program, és jelzi azt, hogy melyik az aktuális oszlop),
- ez a grafikon mindig elfér a képernyő megfelelő részén ($D \leq 16!$).

5. Összegző oszlopdiagram (hisztogram) készítése:

- szemléltesd azt, hogy hányszor volt az első szobában 0, 1, ..., D darab darázs
- nem biztos, hogy ez a grafikon elfér a képernyőn (gondolj például 1000 kísérlet-re), így ilyenkor gondoskodnod kell a megfelelő kicsinyítésről úgy, hogy azért használj ki a képernyő minél nagyobb részét,
- a grafikon megnézése után legyen mód a szimuláció folytatására.

1986. Első forduló

Első-ötödik osztályosok

A feladatok HT-1080Z-re készültek.

Kérjük a kedves tanár kollégákat, hogy a kijavított dolgozatokból csak azokat küldjék meg a versenybizottságnak, amelyek

- a második...kilencedik feladatok közül legalább egynek hibátlan (maximális pontszámú) megoldását tartalmazzák, és
- legalább 22 pontot értek.

Kérjük, statisztikai értékelés céljából azt is jelezzék, hogy hány dolgozatot nem továbbítottak.

Odaadó munkájukat ezúton is köszönjük.

Budapest, 1986. március 14.

1. feladat (11 pont)

1.1-1.6-ig: add meg a helyes válasz(ok) betűjelét!

1.7-1.10-ig: a csoportokban "kakukktojások" vannak: egy-egy fogalom más, mint a többi; add meg ezek betűjelét!

1.1 Mi az asszembler? (1 pont)

- A: egy nyelv
- B: egy program
- C: egy számítógép
- D: egy híres számítástechnikus
- E: egy számítógépgyártó cég

1.2 Mely állítás(ok) igaz(ak) a hexadecimális számrendszerre? (1 pont)

- A: számjegyeit 0..9-cel és A..F-fel jelöljük
- B: alapszáma 16
- C: egy hexadecimális szám mindig H-val kezdődik
- D: két hexadecimális számjeggyel 255-ig lehet elszámolni
- E: két hexadecimális számjeggyel 65535-ig lehet elszámolni

1.3 Mi az EPROM? (1 pont)

- A: elektromosan programozható, írható és olvasható tár
- B: mikrokazetta adattároláshoz
- C: speciális eszközzel törölhető és programozható, csak olvasható tár
- D: elektromosan törölhető tár
- E: mágneslemez adattároláshoz

1.4 Mi a szoftver? (1 pont)

- A: a számítógéphez használható programok összessége
- B: az operációs rendszer
- C: a fordítóprogramok összessége
- D: a számítógép gépi berendezéseinek összessége
- E: a számítógéphez adott gyári programok

1.5 Mely állítás(ok) igaz(ak) a bit fogalmára? (1 pont)

- A: a kettes számrendszer egy számjegye
- B: a számítógép méretének egysége
- C: hexadecimális számjegy
- D: a számítógéppel ábrázolható legkisebb szám
- E: az információ egysége

1.6 Mi a megszakítás (interrupt)? (2 pont)

- A: a számítógép egyes részeinek kikapcsolása
- B: külső jel hatására a futó program végrehajtása felfüggesztődik, és egy másik programrész indul el
- C: külső jel hatására a számítógép előlről kezdi működését
- D: a BASIC program megállítása STOP utasítással
- E: a BASIC program megállítása END utasítással

1.7 (1 pont)

- A: PRINT
- B: INPUT
- C: INKEY\$
- D: PEEK
- E: INP

1.8 (1 pont)
 A: GOSUB
 B: IF
 C: RETURN
 D: ON GOTO
 E: GOTO

1.9 (1 pont)
 A: regiszter
 B: rekesz
 C: jelzőbit
 D: szubrutin
 E: verem

1.10 (1 pont)
 A: LOG
 B: EXP
 C: TAB
 D: TAN
 E: SIN

2. feladat (6 pont)

Zsolti terepasztalt épített. A vonatok és a terepasztal (váltók, jelzők) állapotának lekérdéséhez számítógépet használ. Az alábbi szubrutin meghívásakor az X és Y változóiban olyan, 255-nél nem nagyobb számok vannak, amelyeknek egyes bitjei vagy bitcsoportjai a terepasztal és a vonatok állapotára jellemző információkat hordoznak.

Határozd meg az egyes bitek szerepét!

```

20 PRINT "ELSŐ VONAT";X AND 7;"CMS"
30 PRINT "MÁSODIK VONAT";Y AND 7;"CMS"
40 PRINT :PRINT "VÁLTÓK"
50 FOR I=3 TO 7 :PRINT TAB(3);I-2;".";
60 IF (X AND INT(2^I))/INT(2^I)=1
      THEN PRINT "kitérdő" ELSE PRINT "egyenes"
70 NEXT I
80 PRINT :PRINT "JELZŐK"
90 FOR I=3 TO 7 :PRINT TAB(3);I-2;".";
100 IF (Y AND INT(2^I))/INT(2^I)=1
      THEN PRINT "tilos" ELSE PRINT "szabad"
110 NEXT I
120 RETURN
  
```

3. feladat (5 pont)

Értsd meg az alábbi programot, és válaszolj a kérdésekre!

```

10 INPUT A
20 INPUT N
30 IF N<1 OR N<>INT(N) OR LEN(STR$(INT(A)))+N>10
   THEN PRINT "H" : GOTO 20
40 N=10*N : V=INT(N*A)
50 IF N*A-V>=.5 THEN V=V+1
60 A=V/N

```

3.1 Mit csinál a program?

- A: semmit, hibajelzéssel megáll
- B: kerekíti A-t N tizedesjegyre
- C: kerekíti A-t N+1 tizedesjegyre
- D: levág A-ból N tizedesjegyet
- E: valami mást

3.2 A számolás során lehet-e túlcsondulás?

- A: nem, mert nem működik
- B: igen, ha A túl nagy
- C: igen, ha N túl nagy
- D: soha, mert védve van túlcsondulás ellen
- E: egyéb esetekben igen

4. feladat (4 pont)

Egy szakácskönyvben ezt olvastuk:

***GRÍZGALUSKA KÉSZÍTÉSE**

Hozzávalók: 1 tojás, 5 dkg gríz, só.

Egy tojásfehérjét felverünk habnak. Belekeverjük a sárgáját, csipet sót és 5 dkg grízt. Forró vízben addig főzzük, amíg a grízszemek üvegesek nem lesznek."

Ez a leírás egy algoritmus, amely több részalgoritmusból áll, amelyek további részalgoritmusokból állnak s.í.t. Eleminek tekintett algoritmusokból összetett algoritmusokat úgy hozhatunk létre, hogy 1. felsoroljuk a részalgoritmusait, 2. megismétlünk bizonyos részalgoritmusokat, 3. választunk részalgoritmusok között. Az algoritmusok műveleteket hajtanak végre tárgyakkal, tárgyakon (ezeket a számítástechnikában gyakran adatoknak nevezzük). Algoritmusok és adatok viszonya többféle lehet: 1. Az algoritmuson kívül létező adatokat – az algoritmus szempontjából – külső adatoknak hívjuk. 2. Az algoritmus működése során létrejövő és az algoritmus befejeztével megszűnő adatokat belső adatoknak nevezzük. 3. Adatok lehetnek bemenő vagy kimenő paraméterei egy algoritmusnak.

4.1 Milyen fogalmat takarnak az ételreceptben szereplő nevek?

Fogalom: belső adat (BA), külső adat (KA), bemenő paraméter (BP), kimenő paraméter (KP), algoritmus (A).

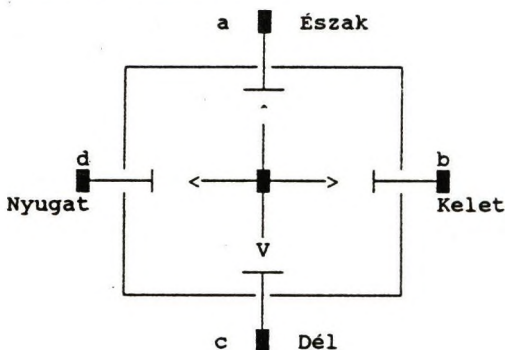
Név: tojás, tojásfehérje, tojás, víz, grízgaluska, üvegesek nem lesznek.

4.2 Mondj példát az ételreceptből ismétlést tartalmazó algoritmusra!

5. feladat (9 pont)

Zolit mérnök barátja ezúttal botkormányral lepte meg. "Rohannom kell!" – mondta köszönés helyett is a mérnök. – "En csináltam négy kapcsolóból, itt van a rajza. A kipróbálásához írtam egy kis programot. Lehet, hogy rossz, nem volt időm futtatni. Sok sikert!"

Segíts Zolinak, magyarázd meg a 'hosszú várakozás' és a 'rövid várakozás' szerepét a programban, és javítsd ki a hibákat!



A botkormány a jelzett négy irányban mozgatható, egyszerre csak egy érintkező záródhat. Az a, b, c és d jelű kimenetek illesztőegységen keresztül csatlakoznak a számítógéphez, állapotukat az INP(60) utasítással lehet megtudni: a zárt érintkező 0-t, a nyitott pedig 1-et jelez az alábbi bitpozíciókban (a 7-4. bitek értéke mindig 1):

7.	6.	5.	4.	3.	2.	1.	0	bit
1	1	1	1	d	c	b	a	

```

10 REM pergésgátolás
20 A = INP(60): IF A = 255 THEN GOTO 20
30 FOR I = 1 TO 40: NEXT I
40 IF INP(60) <> A THEN GOTO 20
50 GOSUB 1000
60 REM hosszú várakozás
70 FOR I = 1 TO 150: NEXT I
80 IF INP(60) <> A THEN GOTO 20
90 GOSUB 1000
100 REM rövid várakozás
110 FOR I = 1 TO 20: NEXT I
120 IF INP(60) <> A THEN GOTO 20
130 GOSUB 1000
140 GOTO 110

1000 REM a kód értelmezése
1010 A = A AND 31
1020 IF A = 14 THEN PRINT "D";: RETURN
1030 IF A = 11 THEN PRINT "É";: RETURN
1040 IF A = 13 THEN PRINT "K";: RETURN
1050 IF A = 8 THEN PRINT "N";: RETURN
1060 RETURN

```


6. feladat (8 pont)

Mi lesz végrehajtás után az A, B, C változók értéke? (A '=' jelölés az értékadást jelöli.)

```
A := 0
B := 1
C := 1
Ciklus amíg B <= N
  A := A + 1
  C := C + 2
  B := B + C
Ciklus vége
```

7. feladat (11 pont)

Az $A(N,N)$ mátrixban egy fekete-fehér képet tárolunk, képpontonként. $A(I,J) = 0$, ha a képpont teljesen fehér; 255, ha fekete; a közbülső értékek a szürke egyenletesen sötétedő árnyalatait jelentik.

Mit tesznek a képpel a következő transzformációk (N páratlan szám)?

- 7.1 100 FOR I = 0 TO N : FOR J = 0 TO N
110 B(I,J) = A(I,J) * A(I,J) / 255
120 NEXT J : NEXT I
- 7.2 100 FOR I = 0 TO N/2 : FOR J = 0 TO N/2
110 C(I,J) = (A(I*2,J*2) + A(I*2+1,J*2) + A(I*2,J*2+1) + A(I*2+1,J*2+1)) / 4
120 NEXT J : NEXT I
- 7.3 100 FOR I = 0 TO N : FOR J = 0 TO N
110 D(I,J) = A(I/3,J/3)
120 NEXT J : NEXT I

8. feladat (6 pont)

Egy programnyelv 3 utasításának jelentése a következő:

REPEAT darabszám [utasítások]

a zárójelbe tett 'utasítások'-at
'darabszám'-szor megismétli.

FORWARD lépésszám

'lépésszám' hosszan vonalat húz az aktuális helyzetből kiindulva az aktuális irányban.

LEFT szög

az aktuális irányt balra, a fokokban megadott 'szög'-gel megváltoztatja.

Itt van egy program ezen a nyelven:

```
REPEAT A [FORWARD B LEFT C]
```

Mi történik, ha az A, B, C változóknak rendre az alábbi értékeket adjuk?

	8.1	8.2	8.3	8.4	8.5
A	3	4	360	3	5
B	100	100	1	10	100
C	120	90	1	240	144

9. feladat (6 pont)

Milyen bemenő adatokkal működik a következő programrészlet a leggyorsabban, ill. a leglassabban ($N > 2$ és állandó)?

```

100 I=N
110 IF I<2 THEN 190
120 CS=0: J=1
130 IF J=I THEN 170
140 IF A(J)>A(J+1) THEN X=A(J):A(J)=A(J+1):A(J+1)=X:CS=J
150 J=J+1
160 GOTO 130
170 I=CS
180 GOTO 110
190 ...

```

1986. Második forduló

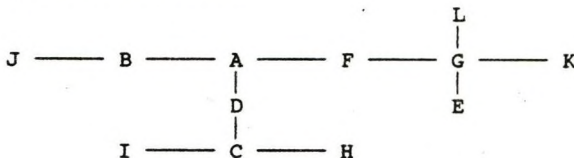
Első-ötödik osztályosok: Újságszállítás

Tervezz és írd programot a Neumann János Számítógéptudományi Társaság tagjainak nyilvántartására, amely a számítástechnikai szakfolyóiratok szétküldésére és a szállítási útvonalak meghatározására szolgál.

Tárolandó adatok:

- név (max. 25 betű; a nevekben ékezetes betű nem szerepel),
- város (számuk max. 16, jelük egy betű az [A,P] tartományban; pl. B város),
- kért újság (vagy a Számítástechnika vagy a Mikroszámítógép Magazin rendelhető meg).

Az újságokat vasúton szállítják. A vasúti összeköttetéseket a felhasználó adja meg. A hálózat városokból mint csomópontokból és ezeket összekötő vonalszakaszokból áll. A hálózat megadásának és tárolásának módját te határozd meg. Egy példa a lehetséges összeköttetésekre:



Megoldandó részfeladatok:

A. A tagok adatainak nyilvántartása:

1. Az adatbázis létrehozása legalább 30 tagra.
2. Új tagok felvétele.
3. Tag törlése.
4. Tag adatainak módosítása.
5. A tagok névsor szerinti listája minden hozzá tartozó adattal.
6. A tagok listája újságonként külön-külön, ezen belül városok szerint ábécé sorrendben.

B. Szállítással kapcsolatos feladatok:

1. A vasúti hálózat modelljének létrehozása a hozzá tartozó kezelőfunkciókkal. Ezek a funkciók a választott beviteli módtól függenek, így ezeket is neked kell meghatároznod.
2. Újságonként külön-külön lista készítése arról, hogy melyik városba mennyi újságot kell küldeni, és ebből hányat kell továbbítani a következő állomás(ok)ra. Feltételezzük, hogy mindkét újságot ugyanott, az "A" városban nyomtatják, és hogy "A" városból minden városba el lehet jutni egy és csakis egy útvonalon.
3. A feladat ugyanaz, mint az előző pontban, de a két újságot bármelyik két különböző városban nyomtathatják, és néhány város több útvonalon is elérhető.

Kiegészítések:

- A listákat a képernyőre kell kiírni.
- Az adatbázisok háttértárolását nem kell megoldani.
- A program a lehető legjobban legyen dokumentálva, feleljen meg az általános formai követelményeknek, és legyen barátságos a felhasználóval.
- A program első sora egy REM sor legyen, és tartalmazza a következő adatokat: név, iskola, helység.
- A programodat két példányban rögzítsd!
- Beadandó a kazetta (vagy lemez) és minden, a megértést segítő dokumentum, vázlat.

1987. Első fordulóElső-ötödik osztályosok

A 10 feladat összpontszáma: 95.

1. feladat (12 pont)

Melyek a bemenő adatok, s milyenek kell lenniük, hogy a következő programrészlet jól működjön? A program egy számsorozatot úgy ír ki, hogy először a benne szereplő negatív értékeket írja ki, majd pedig a nemnegatívakat! Mi a szerepe az F1, F2 változóknak és a B(N) vektornak?

```

110 F1=0 : F2=0 : X=0
120 FOR I= 1 TO N
130 IF A(I)>=0 THEN B(I)=F2 : F2=I
      ELSE B(I)=F1 : F1=I : IF B(I)=0 THEN X=I
140 NEXT I
150 B(X)=F2 : X=F1
160 IF X<>0 THEN PRINT A(X) : X=B(X) : GOTO 160

```

2. feladat (5 pont)

Egy veremautomata a következőképpen működik: ha számot kap a bemenetén, azt egy verembe teszi, ha műveleti jelet, akkor a megfelelő műveletet a verem tetején lévő számokkal elvégzi, majd az eredményt a verembe teszi (pl. A-B a következőképpen írható le az automata nyelvén: A B -). A veremautomata a következő sorozatot kapja (a számokat egymástól és a jelektől szóköz választja el):

$$13 + 254 * 8 - 6 / + -$$

Add meg, mi lesz a verem állapota az egyes jelek (adat vagy művelet) érkezése után, ill. a feldolgozás végén!

3. feladat (12 pont)

Mit csinálnak a következő rekurzív programok (X az angol ábécé betűinek sorozata)?

F(X):

Ha X üres akkor eredmény:=0
 különben Ha (első(X) (A,E,I,O,U)-valamelyike)
 akkor eredmény:=1+F(elsőnkívüli(X))
 különben eredmény:=F(elsőnkívüli(X))

Függvény vége

F(X):

Ha X üres akkor eredmény:=IGAZ
 különben eredmény:=(első(X) (A,E,I,O,U)-valamelyike)
 ES F(elsőnkívüli(X))

Függvény vége

F(X):

Ha elsőnkívüli(X) üres akkor eredmény:=első(X)
 különben Ha első(X)>F(elsőnkívüli(X))
 akkor eredmény:=első(X)
 különben eredmény:=F(elsőnkívüli(X))

Függvény vége

Megjegyzés:

Az első(X) függvény megadja az X karaktersorozat első tagját. Az elsőnkívüli(X) függvény megadja az X karaktersorozat tagjait a 2.-tól a végéig. Az ES a szokásos logikai ES (AND) művelet.

4. feladat (9 pont)

Milyen hibák vannak a következő programban? Adj meg egy olyan bemenő adatot, amelyre helyesen működik!

```
10 INPUT "KÉREK EGY EGÉSZ SZÁMOT! (>2)";N
15 IF N<3 OR N<>INT(N) THEN 10
20 K=0:L=0
30 FOR I=2 TO SQR(N)
40 IF N/I=INT(N/I) THEN K=K+1
50 IF K=1 THEN L=I
60 NEXT I
70 PRINT "VALÓDI OSZTÓK SZÁMA=";K*2
80 PRINT "LEGKISEBB VALÓDI OSZTÓ=";L
90 STOP
```

5. feladat (12 pont)

A következő LOGO nyelvű program, adott x és f értékre, egy ábrát rajzol a képernyőre (az egyes utasításokat szóköz választja el egymástól).

```
FORWARD 2*x RIGHT f FORWARD x LEFT f/2 BACK x LEFT f/2 BACK x
```

Az utasítások jelentése (a LOGO-ban egy teknőcnek nevezett kurzor segítségével rajzolhatunk, amely az aktuális irányban előre vagy hátra képes lépni, valamint jobbra vagy balra tud fordulni):

FORWARD hossz - előre lép és vonalat húz,
 BACK hossz - hátra lép és vonalat húz,

RIGHT fok - jobbra fordul helyben,
LEFT fok - balra fordul helyben.

Mit kell módosítani, ha azt szeretnénk elérni, hogy a kapott ábra az eredeti tükörképe legyen

- A: az eredeti irányban a kezdőponton át húzott egyenesre,
B: az eredeti irányra merőleges, a kezdőponton át húzott egyenesre,
C: a kezdőpontra?

A végrehajtandó utasítások száma nem változhat! (Próbálj többféle megoldást adni!)

6. feladat (5 pont)

A következő programrészletek az A és B egész számok ($B < 0$) hányadosának egészrészét határozzák meg és helyezik el az X változóban. Egyik sem működik tetszőleges egész számokra. Milyen feltételek mellett helyesek az egyes megoldások (a $B < 0$ feltételen kívül)?

A: Osztas:
Ciklus amíg $B > 1$
Ha A páros és B páros akkor $A := A/2 : B := B/2$
Ciklus vége
 $x := A$
Eljárás vége

B: Osztas:
 $x := 0$
Ciklus amíg $\text{abs}(A) > \text{abs}(B)$
 $A := A - B : X := X + 1$
Ciklus vége
Eljárás vége

7. feladat (10 pont)

A következő gépi kódú program az A és a B regiszterben vár egy-egy egész számot. Mely esetben fejeződik be a végrehajtás az IGEN címkénél, és mely esetben a NEM címkénél ($A, B > 0$)?

CIKLUS:

```

CP 0           ; hasonlítsuk össze A-t 0-val
JP Z,IGEN     ; ugrás, ha A=0 volt
JP M,NEM     ; ugrás, ha A<0 volt
BIT 0,A       ; A legkisebb helyértékű bitje 1?
JP NZ,PTLANA ; ugrás, ha 1 volt
BIT 0,B       ; B legkisebb helyértékű bitje 1?
JP NZ,PTLANB ; ugrás, ha 1 volt
SRL A         ; A bitjeinek léptetése egy hellyel jobbra
               ; a legmagasabb helyértékre 0 lép be
SRL B         ; B bitjeinek léptetése egy hellyel jobbra
               ; a legmagasabb helyértékre 0 lép be
JP CIKLUS     ; ugrás a CIKLUS címkére

```

PTLANA:

BIT 0,B ; B legkisebb helyértékű bitje 1?
 JP Z,NEM ; ugrás, ha 0 volt
 SUB B ; A:=A-B
 JP CIKLUS ; ugrás a CIKLUS címkére

PTLANB:

SRL A ; A biteinek léptetése egy helyel jobbra
 ; a legmagasabb helyértékre 0 lép be
 JP CIKLUS ; ugrás a CIKLUS címkére

IGEN: --

--

NEM: --

--

8. feladat (8 pont)

Egy tetszőleges $F(X)$ folytonos függvény zérushelyét keressük E pontossággal az $[A,B]$ intervallumon (azaz $\text{abs}(F(X)) \leq E$ tulajdonságú X -et keresünk). Milyen elégséges feltételeknek tegyen eleget az $F(X)$ függvény az A és a B pontban, hogy az $[A,B]$ intervallumban biztosan legyen legalább egy zérushelye, és a programrészlet ezek közül egyet megtaláljon?

A: $K := (A+B)/2$ Ciklus amíg $\text{abs}(F(K)) > E$ Ha $F(K) < 0$ akkor $A := K$ különben $B := K$ $K := (A+B)/2$

Ciklus vége

Ki: $K, F(K)$ B: $K := A - F(A) * (B - A) / (F(B) - F(A))$ Ciklus amíg $\text{abs}(F(K)) > E$ Ha $F(K) * F(A) < 0$ akkor $B := K$ különben $A := K$ $K := A - F(A) * (B - A) / (F(B) - F(A))$

Ciklus vége

Ki: $K, F(K)$ 9. feladat (7 pont)

Írd át gyorsabb működésűre a következő programrészletet! (A program BASIC nyelvű legyen, és egy sorban csak egy utasítás lehet.) A program H_0 magasságból leeső M tömegű test helyzeti és mozgási energiáját számolja ki méterenként.

100 INPUT M,H0

110 DIM X(H0),Y(H0)

120 G=9.81

130 FOR H=H0 TO 0 STEP -1

140 EH=M*G*H

150 EM=M*G*H0-M*G*H

160 X(H)=EH

170 Y(H)=EM

180 NEXT H

10. feladat (15 pont)

Pisti mikroszámítógépéhez kapott egy "egeret". Ez egy cigarettásdoboz méretű eszköz, amely az asztalon szabadon tologatható, és elmozdítását egy vezetéken keresztül közli a számítógéppel. Az eger állapota az INP(63) utasítással kérdezhető le. Pisti írt egy rövid programot:

```
10 PRINT INP(63);
20 GOTO 10
```

A program elindítása után Pisti várt egy kicsit, majd lassan elmozdította az egeret először jobbra, majd balra, azután előre, végül hátra. Az irányváltások előtt picit mindig várt. A képernyőn a következő számsorozat jelent meg:

```
|<-- az eger áll -->|<----- jobbra mozog ----->|
240 240 240 240 240 242 243 241 240 242 243 241 240 242
|<----- áll ----->|<----- balra mozog ----->|<-- áll -->|
242 242 242 242 240 241 243 242 240 241 243 243 243 243
|<----- előre mozog ----->|<----- áll ----->|
251 255 247 243 251 255 247 243 243 243 243 243
|<----- hátra mozog ----->|<----- áll ----->|
247 255 251 243 247 255 251 243 247 247 247 247 247
```

Pisti ezután a következő programot írta (egyes részeket neked kell megírnod!):

```
10 X=10:Y=10:X0=0:Y0=0
20 E=INP(63):E1=E-16*INT(E/16):Y1=INT(E1/4):X1=E1-4*Y1
30 X9=X0*10+X1
40 IF feltétel1 THEN X=X+1:GOTO 60:REM jobbra
50 IF feltétel2 THEN X=X-1:REM balra
60 Y9=Y0*10+Y1
70 IF feltétel3 THEN Y=Y+1:GOTO 90:REM előre
80 IF feltétel4 THEN Y=Y-1:REM hátra
90 PLOT X,Y:REM kirajzolja az (X,Y) koordinátájú pontot
100 X0=X1:Y0=Y1:GOTO 20
```

- A: Mit jelentenek az egyes bitek?
 B: Mi a szerepe az (X,Y), (X0,Y0), (X1,Y1), (X9,Y9) változó pároknak?
 C: Mit csinál a program, és mi jelenik meg a képernyőn?
 D: Add meg a feltétel1, feltétel2, feltétel3, feltétel4 helyére beírandó, X9-től vagy Y9-től függő logikai kifejezéseket!

1987. Második fordulóElső-ötödik osztályosok: Üzenetközvetítő

A Zabfeldolgozó és Forgalmazó Vállalat – továbbiakban ZAFFO – székházának portáján egy olyan számítógép van, mint előtte. Ez a gép üzenetközvetítőként működik. Minden vezető beosztású dolgozó, valamint a rendszer működéséért felelős programozó a nevével, a 6 osztály többi dolgozója pedig a nevével és osztályának azonosítójával jelentkeznek be. A vállalatnak nincs két azonos nevű dolgozója. A dolgozó kiírhatja a neki szóló leveleket, leveleket küldhet és törölhet. A következő levéltípusok vannak:

Fajta	Címzett	Küldő
körlevél	minden dolgozó	vezérigazgató, ill. az MSZMP, a KISZ és a szakszervezet titkára
körlevél	igazgatók, titkárok	vezérigazgató
körlevél	osztályvezetők	vezérigazgató, igazgatók
körlevél	egy osztály dolgozói	vezető dolgozók
levél	egy dolgozó	bárki

A különböző levéltípusokat a következő szabályok alapján törölhetjük:

- egy olvasás után törölendő,
- adott dátumig mindig közlendő,
- visszavonásig közlendő – a levél küldője vonhatja vissza,
- a levelet kapó törölheti.

A törlési szabályt a levél küldője határozza meg. Nem mindenfajta levélre alkalmazható bármelyik törlési szabály, az értelmetlen kapcsolatokat a programnak ki kell zárnia.

A program funkciói:

- a napi dátum beírása (ezt minden reggel Kiss Attila programozó végezheti el),
- levél küldése,
- az összes levél összes adatának kiírása (ezt csak Kiss Attila programozó kérheti),
- a dolgozónak szóló levelek kiírása,
- a levél törlése (Kiss Attila bármely üzenetet törölhet).

(Javasoljuk, hogy az egyes részfeladatokat ilyen sorrendben oldd meg!)

A ZAFFO vezető dolgozói:

Nagy Lajos	vezérigazgató
Kiss Benedek	gazdasági igazgató
Sipos Albin	műszaki igazgató
Szabados Erika	kereskedelmi igazgató
Kun Ilona	az áruforgalmi osztály vezetője
Forint Ede	a pénzügyi osztály vezetője
Talpas Egon	a könyvelési osztály vezetője
Debreceni Attila	a személyzeti osztály vezetője
Tokos Ibolya	a termelési osztály vezetője
dr. Antal Frigyes	a műszaki osztály vezetője
Koszorú Gyula	MSZMP titkár
Kezes Aranka	KISZ titkár
Petrovics Etelka	szakszervezeti titkár

Készítsd el a program tervét: definiáld az adatszerkezetet és a fontosabb változókat, tervezd meg a program algoritmusát; majd írd meg a feladatot megoldó BASIC programot! A program tagolt, áttekinthető, egyszerű legyen! A programnak a dátum helyességét nem kell ellenőriznie! A háttértáron való tárolást nem kell megoldanod! A levél szövege nem lehet hosszabb, mint ami elfér egy szöveg típusú változóban!

A munka végeztével be kell adni a programtervet, a program listáját kinyomtatva, a programot kazettán vagy lemezen.

A feladat megoldásához bármilyen írásos segédeszköz, ill. bármilyen BASIC bővítés (pl. SIMON'S BASIC) használható.

1988. Első fordulóElső-ötödik osztályosok1. feladat (6 pont)

Az $(x - 10.0^i)^2 = 0.0$ egyenlet gyökeit az

```

a := 1.0;
b := -(10.0**i + 1.0);
c := 10.0**i;
x1 := (-b - SQRT(b**2 - 4.0*a*c)) / (2.0*a);
x2 := (-b + SQRT(b**2 - 4.0*a*c)) / (2.0*a)

```

utasításokkal akarjuk kiszámíttatni az $i = 1, 2, \dots, 25$ értékekre, mégpedig egy olyan számítógépen, amelyen a valós számokat lebegőpontos alakban ábrázolják: a mantissza 24 bites bináris szám (az egyik bit az előjel), a 2-es alapú kitevő pedig 8 bites, ugyancsak bináris szám (az egyik bit itt is az előjel). A programban a $**$ a hatványozás jele.

Ha x_1 -et és x_2 -t kiíratjuk, milyen i értékek esetén nem kapjuk meg a helyes megoldást?

2. feladat (15 pont)

Az alábbi program az m természetes számhoz megkeresi annak a leghosszabb $[i, j]$ zárt intervallumnak a hosszát, amelyre az $i, i+1, \dots, j$ természetes számok összege pontosan m ($1 \leq i \leq j$).

```

hossz := 0;
[1] WHILE m > az első hossz szám összege
    REP hossz := hossz + 10 ENDREP;
[2] WHILE m < az első hossz szám összege
    REP hossz := hossz - 1 ENDREP;
[3] WHILE (m - az első hossz szám összege) MOD hossz <> 0
    REP hossz := hossz - 1 ENDREP.

```

```

az első hossz szám összege:
    IF hossz MOD 2 = 0
    THEN hossz DIV 2 * (hossz+1)
    ELSE (hossz + 1) DIV 2 * hossz
    ENDIF.

```

A programban hossz $(=j-i+1)$ és m egész számokat jelölnek, DIV egészosztás hányadosát, MOD egészosztás maradékát adja eredményül, WHILE ... REP ... ENDREP előlívizsgált ismétlés, az első hossz szám összege a kiszámított értéket átadó (a ún. függvényeljárásokhoz hasonló) algoritmus.

- 2.1. Mi a szerepe az [1]-gyel, [2]-vel, ill. [3]-mal jelölt ismétléseknek?
- 2.2. Hányszor kell végrehajtani a [3] ismétlést 4095, ill. 4096 esetén?
- 2.3. Mutasd meg, hogy a program biztosan befejeződik (azaz nincs benne végtelen ciklus)!
- 2.4. Mutasd meg, hogy ha m 2 nemnegatív hatványa, akkor a maximális intervallumhossz éppen 1!

- 2.5. Hogyan lehetne gyorsabbá tenni a programot a meglévő jó tulajdonságok megőrzésével?

3. feladat (8 pont)

Egy dobozban fekete és fehér kávészemek vannak. Véletlenszerűen kivesszünk két szemet. Ha ezek azonos színűek, akkor helyettük egy feketét rakunk vissza. (Erre a célra korlátlan mennyiségű fekete kávészem áll a rendelkezésünkre.) Ha viszont különböző színűeket húzunk, akkor egy fehéret dobunk vissza a dobozba.

- 3.1. Hogyan változik a fekete, ill. a fehér kávészemek száma a folyamat során?
- 3.2. A kávészemek számának milyen lényeges tulajdonsága nem változik meg az algoritmus végrehajtása közben?
- 3.3. Mitől függ, hogy milyen színű kávészem marad a dobozban?

4. feladat (9 pont)

A Sakál üzletház pénztárgépeihez vonalkódolvasó készülékeket vásárolt. Néhány hónap alatt kiderült, hogy a készülékek nem a nemzetközileg elfogadott vonalkódot, hanem valami mást ismernek fel. Bit Benő elhatározta, hogy megfejti a kódot. A készülék gépkönyvében erről a következőket találta:

- A vonalkódban a bináris 0-kat egyszeres, az 1-eseket kétszeres vastagságú sötét vonal jelöli. A sötét vonalakat legfeljebb kétszeres vastagságú fehér csíkok választják el egymástól.
- Egy-egy árucikk azonosítója 1000 ... 6999 közötti decimális szám; minden számjegyet 4 sötét vonallal kell kódolni. Példák:

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ = 1920,

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ = 6375.

- A kódolvasó ceruzát a papírra kell tenni, és a vonalakra lehetőleg merőleges irányban folyamatos kézmozgással kell végighúzni.
- Az árucikk száma nem függ attól, hogy a kódolvasó ceruzát előről hátrafelé vagy fordítva húzzuk-e végig a kódolt számsoron.

4.1. Mit állapított meg Benő, mi a számjegyek kódja?

4.2. Hogyan ismerhető fel a cikkszám mindkét irányból?

5. feladat (6 pont)

Mit csinál a következő program, és milyen bemenő adatokra működik rosszul (feltéve, hogy a\$-ba számjegyeket, b-be pedig pozitív egész számot írunk)?

```
10 INPUT "Számjegyeket kérek";a$
20 INPUT "Pozitív egész számot kérek";b
30 IF LEN(a$) < 3 OR b >= LEN(a$) THEN 10
40 PRINT MID$(a$,1,1);";";MID$(a$,2,b-2);
50 c$=MID$(a$,b,1)
60 IF MID$(a$,b+1,1) > "5" THEN c$=CHR$(ASC(c$)+1)
70 PRINT c$;"E";LEN(a$)-1
```

MID\$(x\$,p,h) az x\$ karaktorsorozat p-edik pozícióján kezdődő, h hosszúságú karaktorsorozatot állítja elő.

6. feladat (7 pont)

Adott egy 0-kból és 1-esekből álló számsorozat, meghatározandó a csupa 1-esekből álló szakaszok hossza. Pl. a 000111001100 sorozatban van egy 3 és egy 2 hosszúságú 1-es szakasz. Két algoritmust készítettünk a megoldásra, mindkettő az $A(N)$ vektort használja. (Feltesszük, hogy $N > 1$ természetes szám, és az $A(N)$ vektor csak 1 és 0 elemeket tartalmaz.)

```
A: l:=2 : H:=0
   Ciklus amíg l<=N
     Ha A(l-1)=0 és A(l)=1 akkor
       H:=0
       Ciklus amíg A(l)=1
         l:=l+1 : H:=H+1
       Ciklus Vége
       K:=H
     különben
       l:=l+1
   Elágazás vége
   Ciklus vége
```

```
B: l:=1 : H:=A(1)
   Ciklus amíg l<N
     Ha A(l)=0 és A(l+1)=1 akkor H:=0
     Ha A(l+1)=1 akkor H:=H+1
     Ha A(l)=1 és A(l+1)=0 akkor K:=H
     l:=l+1
   Ciklus vége
```

- 6.1. Milyen esetben működnek különbözően, és mi ez a különbség?
- 6.2. Milyen esetben hibás valamelyik, és mi a hiba?

7. feladat (7 pont)

Mit rajzol a következő COMAL nyelvű program, ha feltesszük, hogy a rajzolás során végig a képperryón maradunk?


```

0010 DIM X(5), Y(5)
0020 FOR I:= 1 TO 4 DO
0030 INPUT X(I), Y(I)
0040 ENDFOR I
0050 X(5) := X(1); Y(5) := Y(1)
0060 INPUT A
0070 SETGRAPHIC 0
0080 ALFA := 0
0090 RAJZOLÁS
0100 ALFA := A
0110 RAJZOLÁS
0120 STOP

0200 PROC RAJZOLÁS
0210 PENUP
0220 SETXY X(1) * COS(ALFA) + Y(1) * SIN(ALFA),
        -X(1) * SIN(ALFA) + Y(1) * COS(ALFA)
0230 PENDOWN
0240 FOR I:= 2 TO 5 DO
0250 SETXY X(I) * COS(ALFA) + Y(I) * SIN(ALFA),
        -X(I) * SIN(ALFA) + Y(I) * COS(ALFA)
0260 ENDFOR I
0270 ENDPROC RAJZOLÁS

```

SET x,y jelentése: húzz egyenest az aktuális ponttól (x,y)-ig, majd vedd (x,y)-t az aktuális pontnak. SETGRAPHIC 0 grafikus üzemmódba vált át, PENUP felemeli, PENDOWN pedig leteszi a tollat.

8. feladat (9 pont)

Egy $n \times m$ -es tömbben tároljuk az ország térképét. Minden (x,y) koordinátájú pontban $M(x,y)$ jelenti az adott pont tengetsint feletti magasságát. A következő BASIC programrészt megpróbálja megkeresni az ország legmagasabb pontjának koordinátáit, mindig a legmeredekebb növekedés irányában haladva.

Milyen esetekben nem találja meg, és mi történik ilyenkor?

```

100 x=RND(n):y=RND(m):REM 1 és n, ill. m közötti véletlen egész
110 x0=x:y0=y
120 FOR i=x-1 TO x+1:IF i<1 OR i>n THEN 160
130 FOR j=y-1 TO y+1:IF j<1 OR j>m OR (i=x AND j=y) THEN 150
140 IF M(i,j)>=M(x0,y0) THEN x0=i:y0=j
150 NEXT j
160 NEXT i
170 IF x0<>x OR y0<>y THEN x=x0:y=y0:GOTO 110
180 PRINT x,y

```

9. feladat (8 pont)

Egy láncfalpas játékautóra 2 fényérzékelőt szereltünk, amelyek egymással 10-15 fokos szöget zárnak be. A jármű bal, ill. jobb oldali láncfalpát külön is, egyszerre is működtethetjük. A járműnek adható parancsok a következők:

OUT 10,szám	a szám legkisebb helyértékű bitje a bal oldali, a következő pedig a jobb oldali láncoltp mozgatására használható (1 = elindul, 0 = megáll);
INP(11)	értéke a bal oldali fényérzékelő által mért fényerősség (0-255 közötti egész);
INP(12)	értéke a jobb oldali fényérzékelő által mért fényerősség (0-255 közötti egész).

A fényerősség mért értéke attól függ, hogy milyen messze van az érzékelő a fényforrástól, és milyen szöveget zár be vele. Készítettünk a játékautóhoz egy BASIC programot:

```

10 INPUT e: IF e <= 0 THEN 10
20 x = INP(11) - INP(12)
30 IF x > e THEN 100
40 IF x < -e THEN 200
50 OUT 10,3
60 IF ABS(INP(11) - INP(12)) <= e THEN 60
70 OUT 10,0
80 GOTO 20
100 OUT 10,2
110 IF INP(11) - INP(12) > e THEN 110
120 OUT 10,0
130 GOTO 20
200 OUT 10,1
210 IF INP(12) - INP(11) > e THEN 210
220 OUT 10,0
230 GOTO 20

```

- 9.1. Hogyan mozog a játékautó a padlón, ha az álló fényforrást a fényérzékelőkkel egy magasságban helyezük el?
- 9.2. Mi az e változó szerepe?
- 9.3. Álló, ill. mozgó fényforrás mellett milyen esetben lehetnek gondok a program működésével?

1988. Második forduló

Első-ötödik osztályosok: Értelmező program

Pontszám: 84 pont.

Készíts egy grafikus nyelvhez értelmező programot BASIC vagy PASCAL nyelven! A program legyen képes beolvasni az ezen a nyelven írt programszöveget (forrásprogramot), majd értelmezni és végrehajtani azt! Egy programsorban az utasítást a paraméterétől és egyik utasítást a másiktól szóközzel kelljen elválasztani! A forrásprogram végét a V utasítás jelzi (l. lejjebb), ennek beolvasása indítja el – automatikusan – az értelmezést. A forrásprogram végrehajtása során keletkezett ábra annak lefutása után is maradjon látható mindaddig, amíg a felhasználó nem jelzi (egy tetszőleges billentyű lenyomásával), hogy az értelmező alapállapotba kerülhet, vagyis újabb grafikus program befogadására álljon készen! A forrásprogram elindulása előtt maga az értelmező gondoskodik a képernyő letörléséről!

Az értelmezendő utasítások

és

leírásuk:

M <hossz>
(Menj)

A ceruzát az aktuális helyről az aktuális irányban mozgátsa el <hossz>-nyit! (1 egység legyen a gép legkisebb grafikus egysége!) Ha kell, akkor a mozgás nyomra látszódjon a képernyőn (l. az N, R parancsoknál)!

F <szög>
(Fordulj)

Az aktuális helyen maradván az új haladási irány legyen a régi irány és <szög> összege! (A <szög> előjeles mennyiség, pozitív irány az óra járásának megfelelő irány.)

N
(Nerajzolj)

Az utasítás kiadása után a következő R utasításig a ceruza mozgásának nyoma ne látszódjon!

R
(Rajzolj)

A következő N utasításig lehessen a képernyőn követni a ceruza mozgását!

T
(Törölj)

Hatására törölődjön a képernyő, de a ceruza aktuális helyzete (hely, irány, és hogy rajzol-e vagy sem) ne változzon!

H
(Haza)

Hatására a ceruza kerüljön az alapállapotba, azaz hely := képernyő közepe, irány := függőlegesen felfelé, és rajzoló üzemmódban legyen! (A hazamozgás során természetesen rajzoljon, ha rajzoló üzemmódban van!)

V
(Vége)

Ez az utasítás jelezze a forrásprogram végét az értelmező számára! Ezután a beírt program azonnal hajtódjon végre!

I <szám> (<utasítás-sor>)
(Ismételd)

Ez az utasítás tegye lehetővé, hogy az <utasítás-sor>-ban leírt teendők <szám>-szor végrehajtdjanak!

E <név> (<utasítás-sor>)
(Eljárás)

Ezzel az utasítással tudunk az itt felsorolt utasításokból definiálni <név> névvel egy új, a többivel egyenrangú utasítást! (Ezeket aktivizálni a <név> leírásával lehessen!)

D <név>
(Dobdél)

Ezzel lehessen a már feleslegessé vált, ÁLTALUNK definiált eljárásokat törölni, hogy ezáltal hely szabaduljon fel az újabbak számára!

FONTOS!

- A kész program mellett be kell adni a program tervét (a fontosabb szubrutinok és a változók leírását)!
- Először a vezérlő programrészt írd meg, majd - ajánljuk, hogy - az utasításokat a fenti sorrendben valósítsd meg!
- Bármilyen gépet és bármilyen nyelvet is használsz, a rajzoláshoz a programodban csak az egy pont rajzolásához és a képernyő törléséhez szükséges utasításokat használhatod fel! (Ebből következik, hogy a szakaszrajzoló rutint is neked kell megírnod!) Ha az általad használt gépen vagy nyelvben nincs grafika, akkor a legkisebb egység nálad - természetesen - a betűméret lesz!

Bővítési lehetőségek:

- Ha az eddigi feladatokat megoldottad, akkor próbáld meg úgy módosítani a programodat, hogy jelezze a felhasználónak a szintaktikus hibákat, majd a futás közbeni hibákat is!
- Ha ezt is túl kevésnek találod, akkor az eddigiek elkészítése után nekiláthatsz az utasítások nevének bővítéséhez oly módon, hogy az egybetűs parancs alatt zárjelben levő szavakkal lehessen utasításokat beírni.

Mintaprogramok:

1. M 20 F 60 M 20 F 60 M 20 V
2. T I 10 (R M 20 F 18 N M 20 F 18) V
3. E T E T O (F 45 M 60 F 90 M 60) T E T O V

1989. Első fordulóElső-ötödik osztályosok1. feladat (3 pont)

Milyen számot ír ki a képernyőre a következő program? Indokold meg!

```

10 I=0
20 GOSUB 50
30 PRINT I
40 STOP
50 GOSUB 60
60 GOSUB 70
70 GOSUB 80
80 GOSUB 90
90 GOSUB 100
100 GOSUB 110
110 I=I+1
120 RETURN

```

2. feladat (8 pont)

Hol vannak hibák az alábbi BASIC szubrutinban? Javítsd ki! Milyen állítást fogalmazhatsz meg a változók tartalmáról és a T\$ vektor rendezettségéről a megjelölt helyeken (a helyes változatban)? A szubrutin a T\$() N elemű vektort rendezné növekvően. A DO WHILE ... LOOP ciklus magja mindaddig végrehajtódik, amíg a WHILE mögé írt feltétel teljesül.

1000 REM BESZÚRÁSOS RENDEZÉS:

1010 FOR I = 2 TO N

1020 ISS = T\$(I) : J = I - 1

←-----1.

1030 DO WHILE J >= 1 OR ISS < T\$(J)

1040 T\$(J) = T\$(J+1) : J = J - 1

←-----2.

1050 LOOP

1060 T\$(J) = ISS

←-----3.

1070 NEXT I

1080 RETURN

3. feladat (6 pont)

Mit számít ki az alábbi program, ha az x változó ($0 < x < 1$) legfeljebb 2 tizedes törtjeget tartalmaz? Milyen feltételek teljesülése esetén fejeződik be a végrehajtás? Mit ír ki a program $x=0.3$ esetén?

```

i := 0; w := '0.';
kiír('x := '); beolvas(x);
r[0] := kerekít(x*100);
repeat
  i := i + 1;
  r[i] := r[i-1] * 2;
  if r[i] >= 100 then begin r[i] := r[i]-100; w := w + '1'; end
  else w := w + '0';
j := 0;
while ((r[i] < 0) and (r[j] < r[i]) and (j < i)) do j := j + 1;
until not ((i < 30) and (j = i));
kiír('Végeredmény:', w);

```

4. feladat (6 pont)

Mit csinál az alábbi algoritmus? Az algoritmus egy T(N,2)$ -es táblázattal és a $K1$, $K2$$ változókkal dolgozik. Az index függvény egy szöveg karakterei alapján 1 és N közötti egész számot generál. (Az 'amíg' ciklus akkor áll le, ha az 'amíg' mögé írt feltétel már nem teljesül.)$

Eljárás:

$K := \text{index}(K1\$) : KK := 0$

Ha T(K,1) \neq ""$ akkor

$KK := K$

Ciklus

Ha $K < N$ akkor $K := K + 1$

különben $K := 1$

amíg $KK \neq K$ és T(K,1) \neq ""$

Ciklus vége

Elágazás vége

Ha $KK = K$ akkor $Ki := \text{"baj!!!"}$

különben T(K,1) := K1$: T(K,2) := K2$$$

Eljárás vége.

5. feladat (6 pont)

Az alábbi algoritmus egy utcai automata pénzvisszaadásának menetét írja le úgy, hogy a lehető legkevesebb számú érmét adja vissza. Az $FT()$ vektor írja le, hogy az automata milyen értéket kezel, értéke (20, 10, 5, 2, 1).

Pénzvisszaadás(ÖSSZEG):

Ciklus $i = 1$ -től 5 -ig

Ha $\text{ÖSSZEG} \geq FT(i)$ akkor $DB := \text{INT}(\text{ÖSSZEG} / FT(i))$

$Ki := DB * \text{"db"}; FT(i); \text{" Ft-os"}$

$\text{ÖSSZEG} := \text{ÖSSZEG} - DB * FT(i)$

Elágazás vége

Ciklus vége

Eljárás vége.

Módosítsd a visszaadás algoritmusát, ha azt is tudjuk, hogy az automatának melyik pénzürméből hány darabja van, amit egy $D()$ vektorban tárolunk!

6. feladat (7 pont)

Mit rajzol az alábbi, LOGO nyelven írt program, ha teknőcünk kezdetben a képernyő közepén tartózkodik és felfelé néz? Mi a GVONAL eljárás szerepe? Mit tartalmaz a paramétere?

```
RIGHT 90 ÁBRA 30
```

Az ÁBRA és a GVONAL eljárások LOGO definíciója:

```
TO ÁBRA X
  IF X>9 THEN GVONAL X LEFT 180 ÁBRA X*2/3 LEFT 180 GVONAL X
END
```

```
TO GVONAL Y
  REPEAT 180 [LEFT 1 FORWARD :Y*3.141592654/180]
END
```

Az utasítások jelentése:

RIGHT fok, LEFT fok

FORWARD lépés

REPEAT db [utasítások]

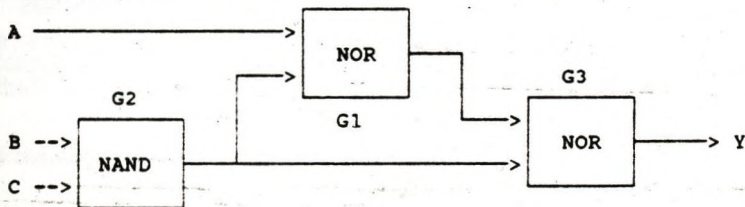
IF

- jobbra, balra fordulás fok fokkal,
- elmozdulás előre lépés hosszan,
- az utasítások megismétlése db-szer,
- a THEN-ág a sor végéig tart.

Az utasításokat egymástól, ill. a paraméterüktől is szökőz választja el. Az eljárások paraméterei elé : -ot kell tenni, az eljárásdefiníció kezdetét TO, végét az END alapszó jelzi.

7. feladat (8 pont)

Egy elektronikus áramköröket gyártó vállalatnak digitális áramkörök hibás kapuit kell megkeresnie. Előzetes tapasztalatok alapján feltételezhető, hogy az ábrán látható áramkörben legfeljebb egy hiba van, és az úgy jelentkezik, hogy valamelyik kapu kimenete a bemeneteitől függetlenül állandóan logikai 0-t ad ki (0-ba ragadt). Ennek megállapításához a G1, G2, G3 kapukból álló áramkör A, B, C bemeneteit a számítógép 63-as kimenetének 0., 1., 2. bitjére kötöttük, az áramkör Y kimenetét pedig a 64-es bemenet 0. bitjére:



Az egyes kapuk igazságtáblázata (NOR = nem vagy, NAND = nem és):

NOR	0	1
0	1	0
1	0	0

NAND	0	1
0	1	1
1	1	0

A következő programmal próbáljuk felfedni a hibákat:

```

10  REM HIBAKERESÉS
20  FOR I = 1 TO 3
30    READ X,Y,H$
40    OUT 63,X          : REM X KIKÜLDÉSE A 63-AS KIMENETRE
50    IF (INP(64) AND 1) <> Y THEN 80 : REM A 64-ES BEMENETEN
                                   NEM Y JÖTT

60  NEXT I
70  H$ = "NINCS 0-BA RAGADT KIMENET"
80  PRINT H$
90  STOP
100 DATA ??,"G1 HIBÁS"
110 DATA ??,"G2 HIBÁS"
120 DATA ??,"G3 HIBÁS"

```

Milyen számok kerüljenek a DATA sorokban levő kérdőjelek helyére? Válaszodat indokold meg!

8. feladat (8 pont)

Egy liftet vezérlő program algoritmusát láthatod az alábbiakban. Gondold végig, hogyan működik ez a lift (ezt nem kell leírnod)! Sorolj fel minél több kritikus (kellemetlen) helyzetet, amelybe a lift utasa kerülhet az algoritmus miatt! A HOVA() vektor 32 elemű. (Hívni a liftet valamelyik emeletről lehet, parancsot adni pedig a fülkében.)

Liftvezérlő program:

SZINT := 0 : UTASÍTÁSSZÁM := 0

Ciklus

Várj amíg nincs HÍVÁS

Jegyezd föl(HÍVÓGOMB)

Ciklus amíg UTASÍTÁSSZÁM > 0

Menj oda(HOVA(UTASÍTÁSSZÁM))

Ajtót nyiss

Várj amíg eltelik 10 másodperc

Ajtót csukj

UTASÍTÁSSZÁM := UTASÍTÁSSZÁM - 1

Ciklus vége

Ciklus vége

Program vége.

Menj oda(x):

Ciklus amíg SZINT <> X

Ha SZINT > X akkor

Menj le eggyel : SZINT := SZINT - 1

különben

Menj föl eggyel : SZINT := SZINT + 1

Elágazás vége

Ha van PARANCS akkor Jegyezd föl(PARANCSGOMB)

Ha van HÍVÁS akkor Jegyezd föl(HÍVÓGOMB)

Ciklus vége

Eljárás vége.

Jegyezd föl(GOMB):

Ciklus l = UTASÍTÁSSZÁM-tól 1-ig -1-esével

HOVA(UTASÍTÁSSZÁM + 1) := HOVA(UTASÍTÁSSZÁM)

Ciklus vége

UTASÍTÁSSZÁM := UTASÍTÁSSZÁM + 1

HOVA(1) := GOMB

Eljárás vége.

(A Várj amíg ..., a Menj ..., az Ajtót ... tevékenységek végrehajtására, a PARANCS-, ill. HÍVÓGOMB érzékelésére, valamint a PARANCS és HÍVÁS logikai értékek beállítására a lift közvetlenül képes, azokat nem kell definiálnunk.)

1989. Második forduló

Első-ötödik osztályosok: Digitális áramkör szimulációja

Készíts programot egy digitális áramkör működésének szimulálására! Az áramkör működésében feltételezzük, hogy minden alapelem kimenetén egységnyi idő múlva jelenik meg a bemeneti jelkombináció hatása, és ez azonnal megjelenik a következő alapelemek bemenetén is. (Azaz az információ a vezetékeken 0, a kapukon egységnyi idő alatt halad át.)

Az áramkör alapelemei a következők lehetnek: INVERTER (NOT), AND, OR, NAND, NOR, EXOR kapuk. (Az inverternek 1, a többi kapunak pedig 2 bemenete van.) Az áramkörben használható kapuk száma maximum 10 lehet. Kezdetben minden kapu bemenetein valamilyen határozott érték (pl. 0) van.

Az áramkör leírását úgy adjuk meg, hogy az alapelemek bemeneteihez, ill. kimenetéhez rendelünk egy csomópont-azonosító számot, és az azonos csomópontokhoz tartozó be-, ill. kimeneteket tekintjük összekötöttnek (l. az ábrát).

A szimulálandó hálózatról, áramköréről tegyük fel, hogy véges időn belül kialakul valamilyen stabil kimeneti jelkombináció (pl. ha N kapu van az áramkörben, akkor ez a korlát legyen N^*N).

Az egyes kapuk működésének leírása:

NOT	
0	1
1	0

OR	0	1
0	0	1
1	1	1

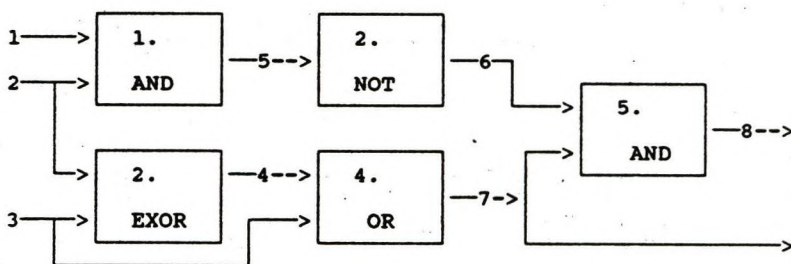
AND	0	1
0	0	0
1	0	1

EXOR	0	1
0	0	1
1	1	0

NOR	0	1
0	1	0
1	0	0

NAND	0	1
0	1	1
1	1	0

Egy példaáramkör (programod kipróbálásához ennél egyszerűbb áramköröket válassz!):



A példaáramkör leírása:

- 1. AND: 1, 2, 5
- 2. EXOR: 2, 3, 4
- 3. NOT: 5, 6
- 4. OR: 4, 3, 7
- 5. AND: 6, 7, 8

Az áramkör bemenetei: 1, 2, 3

Az áramkör kimenetei: 7, 8

Bemeneti kombinációk	Kimenetek
1,1,1	→ 1,0
0,0,0	→ 0,0
0,0,1	→ 1,1

Feladatok:

A: Add meg, hogyan ábrázolod a memóriában az áramkört!

- B: Olvasd be az áramkör leírását tetszőleges, általad választott módon, és ebből építsd fel az áramkört az általad megválasztott adatszerkezet alapján! (Természetesen nem szükséges kirajzolni az áramkört a képernyőre!)
- C: Olvass be bemeneti jelkombinációkat, és az áramkör működésének szimulálásával határozd meg a válaszul kapott kimeneti jeleket! (A kimenetet akkor tekintjük késznek, amikor az áramkör minden alapelemének kimenete változatlan marad.)
- D: Állítsd elő automatikusan az összes bemeneti jelkombinációt, és az áramkör működésének szimulálásával határozd meg a válaszul kapott kimeneti jeleket!
- E: Ellenőrizd, hogy az áramkör leírását helyesen adták-e meg!

A lehetséges hibák:

- egy kapu kimenetét sehova sem kötötték, és az nem is az áramkör kimenete;
- egy kapu bemenetét sehova sem kötötték, és az nem is az áramkör bemenete;
- két (vagy több) kapu kimenetét összekötötték;
- van olyan kapu a hálózatban, amelyet a bemenetről nem lehet elérni;
- az áramkör valamelyik kimenete nem valamelyik kapu kimenete;
- az áramkör valamelyik bemenetét nem kötötték egy kapu bemenetére sem;
- a korlátnak beállított időegységszám után sem alakul ki stabil kimenete az áramkörnek.

1990. Első fordulóElső-második osztályosok1. feladat (4 pont)

A következő algoritmus egy kifejezés zárójelzésének helyességét ellenőrzi. Milyen hibajelenségek tartoznak az egyes HIBAx utasításokhoz?

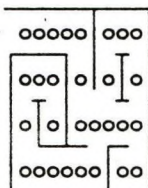
```
Ellenőrzés(X$):
  DB := 0
  Ciklus I = 1-től X$ hosszaiig
    Y$ := X$I. betűje
    Ha Y$ = "(" akkor DB := DB + 1
      különben Ha Y$ = ")" akkor DB := DB - 1
    Ha DB < 0 akkor HIBA1
  Ciklus vége
  Ha DB > 0 akkor HIBA2
Eljárás vége.
```

2. feladat (15 pont)

A következő két algoritmus egy labirintusból vezet ki:

<p>A: Labirintus: Ciklus amíg nem ért ki Balrafordulás Ciklus amíg fal van előtte Jobbrafordulás Ciklus vége Előrelépés Ciklus vége Eljárás vége.</p>	<p>B: Labirintus(X,Y): Ciklus amíg nem ért ki Ha szabad(X,Y-1) akkor Y := Y-1 különben Ha szabad(X-1,Y) akkor X := X-1 Ciklus vége Eljárás vége.</p>
--	---

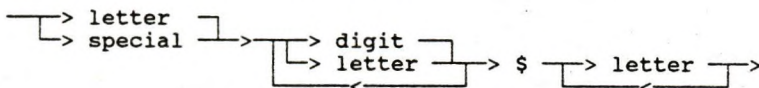
Az ábrán egy labirintus látható, amelynek kijárata a bal felső sarokban van. Másold le két példányban, és jelöld meg benne x-szel azokat a helyeket, amelyekről elindulva előfordulhat, hogy a fenti A és B algoritmusok nem vezetnek ki a labirintusból! Milyen tulajdonságú labirintusokból, ill. pontokból vezet ki biztosan A, ill. B?



3. feladat (10 pont)

Az alábbi szintaxisábrák alapján dönts el, hogy szintaktikailag melyik karaktersorozat helyes és melyik hibás! Magyarázd meg a válaszaidat!

'letter': bármely kisbetű 'a' és 'z' között,
 'digit': bármely számjegy '0' és '9' között,
 'special': az aláhúzás (_) vagy a hátratórt-vonal (\).



- a) april\$
 c) apRIL
 e) \apr\$a

- b) _apr\$a
 d) \apr\$a\$a
 f) apr4\$a

4. feladat (13 pont)

A következő kártyakeverő programban a megjelölt helyen 4 különböző programrész állhat:

```
FOR I := 1 TO 32 DO LAP[I] := I;
FOR I := 1 TO 32 DO
  BEGIN
    (**)
    KIRAJZOL(J)
  END
```

A négy beilleszthető rész a következő:

(*1) J := VÉLETLEN(32);

(*2) REPEAT
 J := VÉLETLEN(32)
 UNTIL LAP[J] > 0;
 LAP[J] := 0;

(*3) K := VÉLETLEN(32);
 J := LAP[K];
 LAP[K] := LAP[I];
 LAP[I] := J;

(*4) K := VÉLETLEN(33-I)+-1;
 J := LAP[K];
 LAP[K] := LAP[I];
 LAP[I] := J;

A program egy 32 lapból álló, megkevert pakli lapjait rajzolja ki. Az egyes kártyalapokat az 1, 2, ..., 32 számok képviselik, a KIRAJZOL eljárás egy lapot rajzol ki a képernyőre, a VÉLETLEN(SZAM) függvény pedig egy 0-nál nagyobb és SZAM-nál nem nagyobb véletlen egész számot állít elő. Szabályosnak akkor számít a keverés, ha minden lap egyszer és csak egyszer fordul elő, mégpedig bármelyik helyen azonos eséllyel.

- A: Melyik algoritmus kever szabályosan és melyik nem?
 B: A szabálytalanul keverő változatokról írd le, hogy mit csinálnak!
 C: Tudsz-e valamit mondani arról, hogy a (*2) változat a FOR-ciklus magjának egy-egy végrehajtása során hányszor hívja meg a VÉLETLEN() függvényt?

5. feladat (7 pont)

Adott a következő BASIC nyelvű program:

```

10 DIM D(300):FOR I = 1 TO 300:D(I) = 0:NEXT I
20 INPUT "HANY SZAM LESZ";N
30 FOR I = 1 TO N
40 INPUT J:IF J <> INT(J) OR J < 1 OR J > 300
   THEN PRINT "HIBAS":GOTO 40
50 D(J) = D(J)+1
60 NEXT I
70 FOR I = 1 TO 300
80 IF D(I) > 0 THEN FOR J = 1 TO D(I):PRINT I:NEXT J
90 NEXT I

```

Mit csinál a program a beadott számokkal (egy mondatban, a lényegét)?

6. feladat (8 pont)

Számítógéppel titkosítottak egy értelmes magyar SZÓ-t. A program lefutása után köztük velünk az előállított KÓD-ot, de a használt KULCS-ot nem.

A kódoláskor használt algoritmus a következő volt:

```

Kódolás:
KÓD(0) := KULCS
Ciklus I = 1-től HOSSZ-ig
  KÓD(I) := SZÓ(I) művelet KÓD(I-1)
Ciklus vége
Eljárás vége.

```

Az 1-től HOSSZ-ig indexelt SZÓ tömb a kódolandó, a 0-tól HOSSZ-ig indexelt KÓD tömb pedig a kódolt szót tartalmazza.

A KÓD (1-től HOSSZ-ig) ez lett: CJNKL. Az alábbi igazságtáblával megadott műveletet a számítógép az operandusokon bitenként végzi el (A és B az operandusok, Q az eredmény).

Művelet:

A	0	1	0	1
B	0	0	1	1
C	0	1	1	0

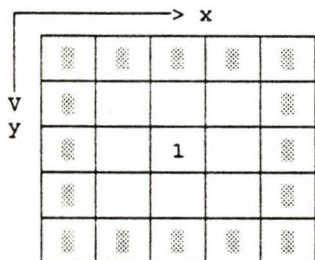
A használható betűk és kódjaik:

- 0000	A 0001	B 0010	C 0011
D 0100	E 0101	F 0110	G 0111
H 1000	I 1001	J 1010	K 1011
L 1100	M 1101	N 1110	O 1111

Mi volt az eredeti, értelmes magyar szó (megjegyzés: az első betű kitalálásához az algoritmus nem ad segítséget), és mi volt a KULCS? Magyarázd meg, hogyan jöttél rá a megoldásra!

Harmadik-ötödik osztályosok1. feladat (15 pont)

Az alábbi algoritmusok a képernyő egy tetszőleges vonallal határolt részének befestésére alkalmasak. Melyik eljárás milyen sorrendben fest be egy bekerített 5*5-ös "négyzetet" a közepéről kiindulva, ha a szélei már be vannak festve (9 befestetlen pontja van)?

**A: Festés(x,y):**

Pontrajzolás(x,y)

Ha üres(x-1,y) akkor Festés(x-1,y)

Ha üres(x,y-1) akkor Festés(x,y-1)

Ha üres(x+1,y) akkor Festés(x+1,y)

Ha üres(x,y+1) akkor Festés(x,y+1)

Eljárás vége.

B: Festés(x,y):

Pontrajzolás(x,y) : Sorba(x,y)

Ciklus amíg Sor nem üres

Sorból(x,y)

Ha üres(x-1,y) akkor Pontrajzolás(x-1,y) : Sorba(x-1,y)

Ha üres(x,y-1) akkor Pontrajzolás(x,y-1) : Sorba(x,y-1)

Ha üres(x+1,y) akkor Pontrajzolás(x+1,y) : Sorba(x+1,y)

Ha üres(x,y+1) akkor Pontrajzolás(x,y+1) : Sorba(x,y+1)

Ciklus vége

Eljárás vége.

C: Festés(x,y):

Pontrajzolás(x,y) : Verembe(x,y)

Ciklus amíg Verem nem üres

Veremből(x,y)

Ha üres(x-1,y) akkor Pontrajzolás(x-1,y) : Verembe(x-1,y)

Ha üres(x,y-1) akkor Pontrajzolás(x,y-1) : Verembe(x,y-1)

Ha üres(x+1,y) akkor Pontrajzolás(x+1,y) : Verembe(x+1,y)

Ha üres(x,y+1) akkor Pontrajzolás(x,y+1) : Verembe(x,y+1)

Ciklus vége

Eljárás vége.

Sor: Olyan adatszerkezet, amelynek a végére lehet új elemet betenni, ill. az elején állót lehet kivenni.

Verem: Olyan adatszerkezet, amelynek a végére lehet új elemet betenni, ill. az utoljára betettet lehet kivenni.

2. feladat (6 pont)

Minek a közelfűtő értékét határozza meg a következő programrészlet?

```
100 S = 0
110 FOR I = 1 TO 1000
120 X = 2 * RND(1) - 1 : Y = 2 * RND(1) - 1
130 IF X * X + Y * Y < 1 THEN S = S + 1
140 NEXT I
150 PRINT S / 250
```

3. feladat (11 pont)

Az a kijelentés, hogy "az egyszerű mondat alanyból és állítmányból áll", formális szabályokkal például így írható le:

- (1) <egyszerű mondat> ::= <alany> <állítmány>
 (2) <egyszerű mondat> ::= <állítmány> <alany>

Megadunk néhány további szabályt is:

- (3) <alany> ::= <névelő> <főnév>
 (4) <alany> ::= <jelző> <főnév>
 (5) <alany> ::= <főnév>
 (6) <állítmány> ::= <jelző>
 (7) <állítmány> ::= <jelző> vagyok
 (8) <névelő> ::= a
 (9) <főnév> ::= Pista
 (10) <főnév> ::= tengeralattjáró
 (11) <főnév> ::= fiú
 (12) <jelző> ::= okos
 (13) <jelző> ::= sárga

E szabályok alapján az alábbi mondatok közül melyek helyesek és melyek hibásak? Válaszodban sorold föl azokat a szabályokat, amelyek alapján az egyes mondatok elfogadhatók!

A: Pista okos

D: sárga a tengeralattjáró

B: Pista okos fiú

E: okos vagyok

C: a tengeralattjáró sárga

4. feladat (10 pont)

A következő algoritmus egy kifejezés zárójelzésének helyességét ellenőrzi. A kifejezésben gömbölyű és szögletes zárójelek is lehetnek. Milyen hibajelenségek tartoznak az egyes HIBAx utasításokhoz?

Ellenőrzés(X\$):

DB1 := 0; DB2 := 0

Ciklus I = 1-től hossz(X\$)-ig

Y\$:= X\$ I. betűje

Ha Y\$ = "(" akkor DB1 := DB1 + 1; Verembe(Y\$)

Ha Y\$ = "]" akkor DB2 := DB2 + 1; Verembe(Y\$)

Ha Y\$ = ")" akkor Ha DB1 = 0 akkor HIBA1

különben DB1 := DB1 - 1; Veremből(Z\$)

Ha Z\$ = "(" akkor HIBA2

Elágazás vége

Ha Y\$ = "]" akkor Ha DB2 = 0 akkor HIBA3

különben DB2 := DB2 - 1; Veremből(Z\$)

Ha Z\$ = ")" akkor HIBA4

Elágazás vége

Ciklus vége

Ha DB1 > 0 akkor HIBA5

Ha DB2 > 0 akkor HIBA6

Eljárás vége.

5. feladat (8 pont)

Számítógéppel titkosítottak egy értelmes magyar SZÓ-t. A program lefutása után közölték velünk az előállított KÓD-ot, de a használt KULCS-ot nem.

A kódoláskor használt algoritmus a következő volt:

Kódolás:

KÓD(0) := KULCS

Ciklus I = 1-től HOSSZ-ig

KÓD(I) := SZÓ(I) művelet KÓD(I-1)

Ciklus vége

Eljárás vége.

Az 1-től HOSSZ-ig indexelt SZÓ tömb a kódolandó, a 0-tól HOSSZ-ig indexelt KÓD tömb pedig a kódolt szót tartalmazza.

A KÓD (1-től HOSSZ-ig) ez lett: HIKJFG. Az alábbi igazságtáblával megadott műveletet a számítógép az operandusokon bitenként végzi el (A és B az operandusok, Q az eredmény).

Művelet:

A	0	1	0	1
B	0	0	1	1
C	0	1	1	0

A használható betűk és kódjaik:

- 0000	A 0001	B 0010	C 0011
D 0100	E 0101	F 0110	G 0111
H 1000	I 1001	J 1010	K 1011
L 1100	M 1101	N 1110	O 1111

Mi volt az eredeti, értelmes magyar szó (megjegyzés: az első betű kitalálásához az algoritmus nem ad segítséget), és mi volt a KULCS? Magyarázd meg, hogyan jöttél rá a megoldásra!

6. feladat (16 pont)

A következő kártyakeverő programban a megjelölt helyen 4 különböző programrész állhat:

```
FOR I := 1 TO 32 DO LAP[I] := I;
FOR I := 1 TO 32 DO
  BEGIN
    (**)
    KIRAJZOL(J)
  END
```

A négy beilleszthető rész a következő:

(*1) J := VÉLETLEN(32);

(*2) REPEAT
J := VÉLETLEN(32)
UNTIL LAP[J] > 0;
LAP[J] := 0;

(*3) K := VÉLETLEN(32);
J := LAP[K];
LAP[K] := LAP[I];
LAP[I] := J;

(*4) IF I < 32
THEN
BEGIN
K := VÉLETLEN(32-I)+1;
J := LAP[K];
LAP[K] := LAP[I];
LAP[I] := J
END
ELSE J := LAP[I];

A program egy 32 lapból álló, megkevert pakli lapjait rajzolja ki. Az egyes kártyalapot az 1, 2, ..., 32 számok képviselik, a KIRAJZOL eljárás egy lapot rajzol ki a képernyőre, a VÉLETLEN(SZÁM) függvény pedig egy 0-nál nagyobb és SZÁM-nál nem nagyobb véletlen egész számot állít elő. Szabályosnak akkor számít a keverés, ha minden lap egyszer és csak egyszer fordul elő, mégpedig bármelyik helyen azonos eséllyel.

A: Melyik algoritmus kever szabályosan, és melyik nem?

B: A szabálytalanul keverő változatokról írd le, hogy miért szabálytalanok!

C: Tudsz-e valamit mondani arról, hogy a szabályosan keverő változatok a FOR-ciklus magjának egy-egy végrehajtása során hányszor hívják meg a VÉLETLEN() függvényt?

7. feladat (7 pont)

Adott a következő BASIC nyelvű program:

```

10 DIM D(300):FOR I= 1 TO 300 :D(I)= 0 :NEXT I
20 INPUT "HANY SZÁM LESZ?":N
30 FOR I= 1 TO N
40 INPUT J:IF J <> INT(J) OR J < 1 OR J > 300
    THEN PRINT "HIBÁS!":GOTO 40
50 D(J)= D(J)+ 1
60 NEXT I
70 FOR I= 1 TO 300
80 IF D(I) > 0 THEN FOR J= 1 TO D(I):PRINT I: NEXT J
90 NEXT I

```

Mit csinál a program a beadott számokkal (egy mondatban, a lényeget)?

8. feladat (14 pont)

A FONTOSKODÓ vállalat nyilvántartásában szerepel ez a 2 táblázat:

alkalmazottak					
azon	név	munkakör	főnöke	fizu	oszt
544	Álmos	szállító	545	7000	2
545	Éber	osztvez	550	9800	2
546	Élő	osztvez	550	15000	3
547	Dolgos	munkás	546	11000	3
550	Főfő	igazgató	NULL	20000	4
551	Buzgó	munkavéd	550	10500	2

osztályok		
kód	osztnev	hely
2	szállitás	Pécs
3	termelés	Pécs
4	igazgatás	Győr

Lekérdezésükre használható a SELECT utasítás:

```

SELECT oszlopnév, oszlopnév, ...
FROM táblázatnév, táblázatnév, ...
WHERE logikai kifejezés;

```

A logikai kifejezésben állhat állandó (pl. "Pécs"), oszlopnév (pl. hely) vagy zárójelben egy újabb, beágyazott SELECT utasítás.

A lekérdezés azokat a sorokat (két táblázat esetén sorpárokat) választja ki a táblázat(ok)ból, amelyekre a logikai kifejezés teljesül. Minden kiválasztott sorból, ill. sorpárból csak a megnevezett oszlopokba eső adatokat kapjuk eredményül. A nem beágyazott SELECT utasítás eredménye kiíródik a képernyőre. Például

SELECT osztznév FROM osztályok WHERE hely = "Pécs" OR hely = "Győr";
 kiírja a pécsi és a győri osztályok nevét.

A: Milyen parancs írja ki

A1: a 10000-nél többet kereső osztályvezetők nevét és fizetését?

A2: a Pécssett dolgozók nevét és osztályok nevét?

B: Mit ír ki az alábbi SELECT parancs?

```
SELECT név
FROM alkalmazottak
WHERE főnöke = (SELECT azon
FROM alkalmazottak
WHERE név = "Főfő");
```

+1 feladat

A +1 feladat megoldása nem feltétele a döntőbe jutásnak! Csak akkor fogj neki, ha már nincs más teendő! A megoldást a versenybizottság értékeli, de csak két esetben veszi figyelembe: ha holtverseny alakul ki a döntőbe jutásért, vagy ha külföldi versenyen való részvételre adódik lehetőség.

Sok sikert! Good luck! Viel Erfolg! Bon courage!

Consider the following Pascal declarations:

```
CONST n1 = ??; (* length of row #1 *)
      n2 = ??; (* length of row #2 *)
TYPE r1 = ARRAY [0..n1-1] OF INTEGER;
      r2 = ARRAY [0..n2-1] OF INTEGER;
```

Write a function or procedure (preferably in Pascal)

```
FUNCTION distinct (a1:r1, a2:r2): INTEGER
```

where both arrays 'a1' and 'a2' contain integers in strictly monotonously increasing order, i.e. neither of them contains duplicates! The function or procedure should return the number of distinct elements: those that appear in one array but not in both.

Names, comments and explanations are preferred in English but in Hungarian are also accepted.

Siehe die folgenden Pascal-Deklarationen:

```
CONST n1 = ??; (* Länge der Reihe #1 *)
      n2 = ??; (* Länge der Reihe #2 *)
TYPE r1 = ARRAY [0..n1-1] OF INTEGER;
      r2 = ARRAY [0..n2-1] OF INTEGER;
```

Schreibe eine Funktion oder Prozedur (bevorzugt in Pascal)

```
FUNCTION verschiedene (a1:r1, a2:r2): INTEGER
```

in der die beiden Blöcke 'a1' und 'a2' ganze Zahlen in streng monoton wachsender Reihenfolge enthalten, das heißt, keiner enthält Duplikate! Die Funktion oder Prozedur soll die Anzahl der unterschiedlichen Werte zurückgeben: also jener Werte, die in einem der Blöcke, aber nicht in beiden Blöcken vorkommen.

Namen, Kommentare und Erklärungen sind in deutsch bevorzugt, aber werden auch in ungarisch angenommen.

Considérez les déclarations Pascal suivantes:

```
CONST n1 = ??; (* longueur de ligne #1 *)
      n2 = ??; (* longueur de ligne #2 *)
TYPE r1 = ARRAY [0..n1-1] OF INTEGER;
      r2 = ARRAY [0..n2-1] OF INTEGER;
```

Écrivez une fonction ou procédure (de préférence en Pascal)

```
FUNCTION distinct (a1:r1, a2:r2): INTEGER
```

ou tous les deux blocs 'a1' et 'a2' contiennent des nombres entiers en ordre croissant rigoureusement monotone, c'est-à-dire qu'aucun ne contienne de duplicata! La fonction ou procédure doit rendre le nombre des éléments différents: ceux qui n'apparaissent que dans un seul bloc.

Noms, commentaires et explications sont préférés en français mais acceptés aussi en hongrois.

(Megjegyzés: A versenyzők a feladat szövegét oroszul is megkapták. Az orosz változatot a cirill karakterkészlet hiánya miatt ebben a feladatgyűjtemény jelen kiadásában nem közöljük.)

1990. Második forduló

Első-második osztályosok: Metróállomás

Készíts programot egy metróállomás (amely nem végállomás) forgalmának szimulálására! A feladatot négy, egyre bővülő lépésben oldd meg! (Beadni csak a végső változatot kell.)

A program kövesse nyomon a metróállomáson történő eseményeket, avatkozzon közbe, ha szükséges, és jelezze ki az egyes helyeken (lépcsők előtt, lépcsőkön, peronon, szerelvényekben) várakozók számát, a lépcsők állapotát, a beavatkozásokat! A működési paramétereket (TA, TE, ML stb.) a program a billentyűzetről olvassa be!

A feladat:

A metrószerelvények két irányba mehetnek, az utasok egy közös peronról szállnak be, ill. a metrókocsikból ide szállnak ki (mozgólépcső ebben a részfeladatban még nincs). A peronon, ill. a metrókocsikban tetszőleges számú utas tartózkodhat.

Megoldandó részfeladatok:

1. Az állomásra véletlenszerűen érkeznek az utasok, mindegyikük a két lehetséges irány valamelyikébe szeretne menni. Az irányt érkezéskor véletlenszerűen kell meghatároznia a programnak.
2. Az érkezők beállnak a megfelelő irányba menő metróra várók közé.
3. A metrószerelvények TE időközönként érkeznek (a két irányból nem feltétlenül egyszerre), TA ideig állnak az állomáson. TE, TA az időegység (l. alább) egész számú többszöröse.
4. Az érkező metrószerelvényben véletlenszerűen meghatározott számú utas van, és ezek egy része száll ki.

5. A metrószerelevény minden várakozót elvisz, aki abba az irányba akar menni. (A TA idő elég a be-, ill. kiszállókknak.)
6. A kiszállók elhagyják a peront.

B. feladat:

Az állomáson 2 mozgólépcső van, az egyik felfelé, a másik lefelé viszi az utasokat. Az utasok fent egy váróterembe érkeznek, a mozgólépcsők a várótermet kötik össze a lenti peronnal. Az előzőkhöz hasonlóan a váróteremben is tetszőleges számú utas tartózkodhat. A lépcső ML hosszúságú (azaz 2*ML utas fér rá).

Megoldandó részfeladatok:

7. Az állomásra érkezők a lefelé menő mozgólépcsőnél sorban állnak.
8. A lépcsőre egy időegység alatt maximum 2 ember léphet. Ha van várakozó utas, az rá is lép a lépcsőre.
9. A lépcső egy időegység alatt egy lépcsőfoknyit halad lefelé (ill. a másik lépcső felfelé).
10. A kiszállók beállnak a felfelé menő lépcsőnél várakozók sorába.
11. Aki felért, az elhagyja a metróállomás területét.

C. feladat:

Az állomás helyiségeinek befogadóképessége adott, a be- és kiszállás időt vesz igénybe. A peronon egyszerre PDB ember fér el, a fenti váróteremben pedig VDB. A felfelé menő lépcső külön kijárást vezet, így a kifelé igyekvők nem növelik meg a fenti váróterem telítettségét. Egy metrószerelevényben maximum MDB ember tartózkodhat.

Megoldandó részfeladatok:

12. Ha a fenti váróterem megtelt, akkor az újonnan érkezők elmennek (pl. más járművel utaznak).
13. A metrószerelevény annyi utast visz el, amennyi még belefér a kocsikba (a többiek megvárják a következő szerelevényt).
14. Ha a peron a lefelé menők miatt megtelt, akkor a szimuláció álljon le!
15. Egy időegység alatt a metrószerelevényből SZ számú utas tud ki-, ill. beszállni. Amíg a kiszállók nem szálltak ki, addig senki sem száll be.
16. Ha a peronon nem férnének el a kiszállók, akkor a kocsikban maradnak.

D. feladat:

Az állomáson 3 mozgólépcső van, amelyekből kettő mindig működik, a harmadik pedig bekapcsolható valamelyik irányban, ha szükséges, ill. kikapcsolható, ha már nincs rá szükség.

Megoldandó részfeladatok:

17. A program felhasználója bármikor kezdeményezhesse a harmadik mozgólépcső elindítását, ill. leállítását (vészleállítás, esti kikapcsolás).
18. A program szükség esetén automatikusan indítsa el a harmadik lépcsőt (ha a peronon vagy a váróteremben már sok utas tartózkodik).
19. Ha a peron lent tele van, akkor a program állítsa le egy időre a lefelé menő lépcső(ke)t! (A 14. pont emiatt érvényét veszti.)

Értékelési szempontok:

1. A programban megvalósított funkciók megbízható működése.
2. A megvalósított funkciók száma, sokrétűsége.
3. A program belső szerkezete: logikus tagolás, általános célú eljárások, bővíthetőség stb.

4. A program olvashatósága: beszédes elnevezések, áttekinthető tördelés, magyarázatok stb.

Harmadik-ötödik osztályosok: Szóelválasztás

Készíts programot, amely a billentyűzetről vagy egy szöveges állományból (file-ból) bevitt, szóköz vagy újsor karakterekkel tagolt, helyes magyar szavakat kötőjelekkel elválasztva (szótagolva) ír ki a képernyőre! A programnak csak az alábbi elválasztási szabályokat kell betartania; ne találj ki új szabályokat! A felsoroltak közül próbálj meg minél többet alkalmazni, de úgy, hogy a program működőképes legyen!

A program alapváltozata minden beolvasott szót külön sorban írjon ki a képernyőre; ha több lehetséges elválasztás közül nem tudja kiválasztani a helyeset, akkor egymás után írja ki az összes változatot! Ha van időd, fejleszd tovább a kezelői felületet úgy, hogy a beolvasott szöveg egyetlen változatban a képernyő tetején jelenjen meg, több lehetséges elválasztás esetén pedig a képernyő alján ajánlja fel kiválasztásra az egyes változatokat! Pl. fő-lül, föl-ül, me-gint, meg-int.

A bevitt szövegben az elválasztás helyét egyenlőségjellel (=) megjelölhetjük; ilyenkor a szót alkotó szórészeket mint önálló szavakat kell elválasztani a szótagolás szabályai szerint.

A program az elválasztás szempontjából ne tegyen különbséget kis- és nagybetűk között!

Az ékezetes magánhangzók jelölése: a', e', i', o', o", u', u", u".

Az egyjegyű hosszú mássalhangzók: bb, cc, dd, ff, gg, hh stb.

A többjegyű rövid mássalhangzók: cs, dz, dzs, gy, ly, ny, sz, ty, zs.

Az (egyszerűsített kettőzött) többjegyű hosszú mássalhangzók (ccs, ddz, ddzs, ggy, lly, nny, ssz, tty, zzs).

Elválasztási szabályok (Forrás: A magyar helyesírás szabályai, 1984.)

Az egyszerű szavak elválasztásának alapja a szótagolás.

1. Minden szó annyi szótagú, ahány magánhangzó van benne. Magánhangzó önmagában is alkothat szótagot. A szótag vagy magánhangzóval kezdődik, vagy -- az elsőt kivéve -- egyetlen rövid mássalhangzóval. Az egyetlen magánhangzót tartalmazó, valamint a háromnál nem több hangból álló szavakat nem választjuk el. Pl. bál, György, sztrájk, te-ker-vé-nyes, trom-bi-ta (de: Leó, szia, apu).
2. A két magánhangzó között lévő egyetlen mássalhangzót -- akár egy-, akár többjegyű a betűje -- a következő szótagba visszük át. Pl. be-tű, ba-tyu, bo-dza, apá-mé, né-gyet, bri-dzsel.
3. A két magánhangzó között lévő kétféle rövid mássalhangzót jelölő (azaz nem kettőzött) betűk közül -- akár egy-, akár többjegyűek -- az elsőt az előző szótagban hagyjuk, a másodikat pedig átvisszük a következőbe. Így járunk el akkor is, ha a két mássalhangzó közül az első hosszú (azaz kettőzött betű jelöli). Pl. am-per, fosz-fát, mor-zsa, ta-risz-nya, tal-pa-lat-nyi, bron-zot, töl-gyes, hall-gat, könny-től.
4. Ha a két magánhangzó közötti mássalhangzó hosszú (azaz kettőzött betű jelöli), ennek egyik jegyét az előző szótagban hagyjuk, másik jegyét pedig átvisszük a következőbe. Pl. ber-reg, mil-li-mé-ter, em-ber-rel, hal-lak, job-ban, víz-zé.
5. Ha a két magánhangzó közötti hosszú mássalhangzót egyszerűsített kettőzött többjegyű betű jelöli, ennek elválasztásakor mind az előző szótag végén, mind a következő szótag elején ki kell írni a teljes rövid megfelelőjét. Pl. meny-nyi, ősz-sze, ágy-gyal, ész-szerű, meg-yel.

6. Ha két magánhangzó között kettőnél több mássalhangzót jelölő betű van, csak az utolsó kerül a következő szótagba. Pl. cent-rum, ost-rom, nyolc-kor, bors-sal, vonz-za, kulcs-csal, metsz-szük, edz-dze, bridzs-dzsel.
7. A ch kétjegyű ugyan, de sokszor egyetlen hangot jelöl, az x pedig két hang jele, de egyetlen jegyű, s így mindkettő egy betűnek számít az elválasztáskor. Ugy tekintjük őket, mint a rövid mássalhangzókat jelölő magyar betűket. Pl. or-chi-dea, pszicho-ló-gus, pra-xis, tech-ni-ka (de: srác-hoz).
8. A ch és x végződésű szavak -val, -vel és -vá, -vé ragos alakjait a következő példák szerint kell elválasztani: pech-hel, bő-rax-szá, Bach-hal.

Az igekötős szavak és a leg-képzős melléknévek kivételével összetett szavakkal programnak nem kell foglalkoznia. Ha a feldolgozandó szövegben összetett szó fordul elő, akkor az összetétel helyét egyenlőségjellel jelöljük meg. Az összetett szavakat alkotó tagok már a szótagolás szabályai szerint választandók el.

9. Ha a kötőjellel írt szóösszetételt a szóhatáron szakítjuk meg, a kötőjelet csak egyszer kell kitenni. Pl. égre-földre, dúl-fúl.
10. A felsőfokú melléknévek leg- képzője önálló szótag. Pl. leg-jobb, leg-job-ban, leg-in-kább (de: le-gen-da).
11. A mássalhangzóra végződő igekötő önálló szótagképző: agyon-, át-, benn-, el-, el-len-, fel-, föl-, fe-lül-, fö-lül-, fenn-, fön-n-, ke-resz-tül-, kü-lön-, meg-, széj-jel-, szét-, to-vább-, túl-, vé-gig-. Pl. meg-int, fel-pró-bál.
12. Külön vizsgálandók a magánhangzóra végződő igekötők (ab-ba-, alá-, be-, be-le-, egy-be-, elő-, elő-re-, fél-be-, fél-re-, hát-ra-, ha-za-, hely-re-, hoz-zá-, ide-, ket-té-, ki-, köz-be-, köz-re-, le-, mel-lé-, ne-ki-, oda-, ősz-sze-, rá-, raj-ta-, szem-be-, vég-be-, visz-sza-), ha utánuk két vagy több mássalhangzó áll. Pl. ki-pró-bál, be-ska-tu-lyáz (de: ber-zen-ke-dik).
13. Külön vizsgálandók az alábbi képzőkre, ill. toldalékokra végződő szavak: -ság, -ség, -szerű, -szer, -szor, -szőr. Pl. ma-lac-ság, köz-ség, száz-szor (de ésszerű ész-sze-rű).

Eszétikai kívánalmak:

14. Az összetett szavak összetételi határának kivételével két magánhangzó közé nem szabad elválasztójelet tenni. Pl. dió-nyi, kia-bál (de: ki-e-mel).
15. Szó elején, ill. végén álló magánhangzót nem szabad leválasztani. Pl. ele-mó-zsia, idiill, ké-mia, ké-miai.
16. Szó elején álló magánhangzócsoporthoz nem szabad leválasztani. Pl. Euró-pa, aero-nau-ta.

Értékelési szempontok:

1. A programban megvalósított funkciók megbízható működése.
2. A megvalósított funkciók száma, sokrétűsége.
3. A program belső szerkezete: logikus tagolás, általános célú eljárások, bővíthetőség stb.
4. A program olvashatósága: beszédes elnevezések, áttekinthető tördelés, magyarázatok stb.

1991. Első fordulóElső-második osztályosok1. feladat (5 pont)

Tekintsük az alábbi algoritmust, amelynek bemenete egy szó, kimenete pedig a "JÓ!" vagy a "ROSSZI" felkiáltás.

A VEREMBE és VEREMBŐL eljárások egy veremtár kezelésére szolgálnak. Az ÜRES eljárással megtudható, hogy a verem üres-e. A verem kezdetben üres.

Az OLVAS eljárás a bemenet következő betűjét olvassa be, a VANMÉG eljárás pedig megvizsgálja, hogy van-e a szónak további betűje.

Ha VANMÉG akkor OLVAS (betű) : VEREMBE (betű)

Ciklus amiíg (VANMÉG és nem ÜRES)

 OLVAS (betű)

 VEREMBŐL (felső)

 Ha (betű < felső) akkor VEREMBE (felső) : VEREMBE (betű)

Ciklus vége

Ha (VANMÉG vagy nem ÜRES) akkor "ROSSZI" egyébként "JÓ!"

Mely szavakra ad "JÓ!" és melyekre "ROSSZI" választ az algoritmus az alábbiak közül?

A. bab B. kiiktat C. odaado D. errearra E. kikerrearrakik

2. feladat (8 pont)

Furcsa számítógépünk a következő formában várja tőlünk a programot. Először meg kell adni az alapismereteket. Másodsor föl kell sorolni azokat a következtetési szabályokat, amelyek alkalmazásával a problémát megoldhatónak gondoljuk.

Alapismeretek például:

1. anyja(Szilágyi Erzsébet, Mátyás).
Ez azt jelenti, hogy Szilágyi Erzsébet anyja Mátyásnak.
2. apja(Hunyadi János, Mátyás).
Ez azt jelenti, hogy Hunyadi János apja Mátyásnak.

Szabályok például:

1. szülője(x,y) HA anyja(x,y) VAGY apja(x,y).
2. nagyszülője(x,y) HA szülője(x,z) ÉS szülője(z,y).
Ez magyarul a következőt jelenti: akkor nagyszülője x y-nak, ha van olyan z személy, akinek x a szülője és ő (azaz z) szülője y-nak.

A logikai kifejezéseket balról jobbra haladva értékeli ki a gép. A kiértékelést abba hagyja, ha a részeredmény alapján már eldönthető a teljes kifejezés értéke.

A: Milyen rokonsági kapcsolatot határoznak meg a következő szabályok?

a. rokon1(x,y) HA szülője(z,x) ÉS szülője(z,y) ÉS $x \diamond y$.

b. rokon2(x,y) HA apja(x,z) ÉS szülője(z,y).

- B: Írd meg a következő relációkapcsolatokat leíró szabályokat!
 a. anyaiNagyszülője(x,y) HA ... {azaz x anyai nagyszülője y-nak}
 b. szülőpár(x,y) HA ... {azaz akiknek közös gyermekük van}

3. feladat (8 pont)

Egy palacsintasütő és egy palacsintaevő ember számára készítettünk egy-egy algoritmust. Úgy tervezzük, hogy a két ember ezeket az algoritmusokat egyszerre – azaz egymással párhuzamosan – hajtja végre. Egy közös tárolót (tányért) használnak, amelyen egyszerre csak egy palacsinta fér el. Egymással nem beszélhetnek, csupán egy-egy, a vasúttállomásokon alkalmazottakhoz hasonló szemaforral jelezhetnek egymásnak. Ehhez a következő utasításokat használhatják fel:

jelezz(SZ) szabadra állítja az SZ szemafort,
 várj(SZ) várakozik, amíg az SZ szemafor nem jelez szabadot, majd ismét tilosra állítja SZ-t, és abbahagyja a várakozást.

Sütő:

Ciklus
 Süss egy palacsintát!
 Várj(ÜRES A TÁNYÉR)
 Tedd a palacsintát
 a tányérra!
 Jelezz(EHETSZ)

Ciklus vége
 Eljárás vége.

Evő:

Jelezz(ÜRES A TÁNYÉR)
 Ciklus

Ciklus vége
 Eljárás vége.

A * helyébe négy utasítást teszünk, különféle sorrendben. Add meg, hogy mely megoldások hibásak és miért! Ha több helyes megoldás is van, vizsgáld meg, hogy melyik mennyire hatékony!

A: Várj(EHETSZ)
 Vedd fel a palacsintát!
 Jelezz(ÜRES A TÁNYÉR)
 Edd meg!

B: Vedd fel a palacsintát!
 Várj(EHETSZ)
 Jelezz(ÜRES A TÁNYÉR)
 Edd meg!

C: Várj(EHETSZ)
 Jelezz(ÜRES A TÁNYÉR)
 Vedd fel a palacsintát!
 Edd meg!

D: Várj(EHETSZ)
 Vedd fel a palacsintát!
 Edd meg!
 Jelezz(ÜRES A TÁNYÉR)

4. feladat (12 pont)

Pap Jancsi fogad Tasziló barátjával, hogy a padláson heverő lyukszalagok egyikén sem fordul elő a "LILLA" szó. De hogy biztos legyen a dolgában, programot ír a lyukszalag átalakítására.

Az alábbi BASIC nyelvű programban az OLVAS(K\$) parancs az 1-es számú lyukszalag következő karakterét olvassa be K\$-ba, az IR(L\$) parancs pedig a 2-es számú lyukszalagra írja ki az L\$ tartalmát. A szalagok végét a "\$" jel jelzi. (Mivel Pap Jancsi még nem nagyon tud programozni, programja hibajelzéssel fog leállni a szalag végén.)

Pap Jancsi minden szalagot átmásol a programmal, majd a már átalakított szalagokat is újra meg újra átfuttatja, amíg csak változást tapasztal. Végül minden szalag eredeti helyett annak utolsó változatát rakja vissza a padlásra.

10 OLVAS(K\$)

15 IF K\$ <> "L" THEN IR(K\$): GOTO 10 ELSE OLVAS(K\$)

20 IF K\$ <> "I" THEN IR("L"): GOTO 15 ELSE OLVAS(K\$)

30 IF K\$ <> "L" THEN IR("LI"): GOTO 15 ELSE OLVAS(K\$)

40 IF K\$ <> "L" THEN IR("LIL"): GOTO 15 ELSE OLVAS(K\$)

50 IF K\$ <> "A" THEN IR("LILL"): GOTO 15 ELSE GOTO 10

60 END

Tasziló, aki megneszelte a turpisságot, már előre dörzsöli a kezét. Jogosan, mert a program valóban rossz.

- A: Tasziló, hogy nyilvánvalóvá tegye diadalát, saját készítésű lyukszalagot dug el Pap Jancsi padlásán. Mi legyen ezen a szalagon, hogy a program biztosan felsüljön? (Azaz bennhagyjon legalább egy LILLA-t)
- B: Pap Jancsi közben maga is rájön a hibára, és kijavítja a programot. Két sorban végez apró módosítást, a többi soron nem változtat, és új sort sem szűr be. Add meg ezt a két sort kijavítva!

5. feladat (11 pont)

A kompatibilis programozóhoz címzett mulatóban egy játékgép áll. A gép tetején egy piros, egy zöld és egy kék lámpa van, amelyek közül mindig csak egy ég; kezdetben a piros. A gépbe "1" és "0" felíratú zsetonokat lehet bedobálni. A gépen van még egy nyomógomb, amely csak akkor nyomható meg, amikor a piros lámpa ég; ilyenkor a gép annyi forintot fizet, amennyi az a 2-es számrendszerbeli szám, amelyet az addig bedobott zsetonokon lévő számjegyek a bedobás sorrendjében balról jobbra sorbaállítva kiadnak (pl. 1,1,0 bedobása 6 Ft-ot ér). A zsetonokat a kasszánál darabonként 10 forintért árulják. A gépbe azonban csak 9 zseton fér. Ha 9 zseton bedobása után a zöld vagy a kék lámpa ég, a pénzünk odaveszett.

Só Sájó szerint az, hogy legközelebb melyik lámpa gyullad ki, csak attól függ, hogy éppen melyik lámpa ég, és hogy milyen zsetont dobunk be. Tapasztalatait a következő táblázatban összegezte:

		Melyik lámpa gyullad ki:			
		P	Z	K	
Melyik lámpa ég:	→				
Milyen zsetont dobunk be:	→	0	p	k	←
	→	1	z	p	←

Serte Petra szerint a gép csak 9-cel osztható összegeket fizet ki. De Petra nem igazán ismeri a gépet, és lehet, hogy téved.

- A. Kapunk-e a géptől pénzt a következő zsetonsorozatok bedobásával?
- 1,1,0
 - 0,1,1,1
 - 1,0,1,0,0
 - 1,0,1,0,1
- B. Igaza van-e Petrának? Ha nincs, akkor mi jellemzi a játékgép által kifizetett összegeket?
- C. Maximum mennyit fizet a gép egyszerre, egy játékban?

- D. Legalább hány zsetont kell bedobnod ahhoz, hogy a játék nyereséges legyen? (Hiszen a zsetonok is pénzbe kerülnek.)

6. feladat (5 pont)

Itt van egy programrészlet:

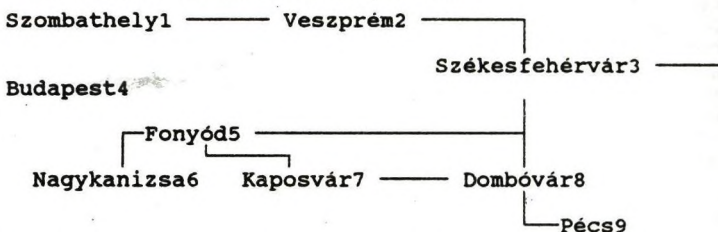
```
Függvényeljárás REJTVÉNY(A, B)
  Ha A > B akkor REJTVÉNY := -REJTVÉNY(B, A)
  egyébként ha A = 0 akkor REJTVÉNY := B
  egyébként REJTVÉNY := REJTVÉNY(A-1, B-1)
Függvényeljárás vége.
```

Mit adnak eredményül a következő függvényeljárás-hívások?

- A. REJTVÉNY(3,12) ?
 B. REJTVÉNY(4,6) + REJTVÉNY(6,4) ?
 C. REJTVÉNY(X,Y), ahol X és Y tetszőleges nemnegatív számok?

7. feladat (7 pont)

Egy vasúti menetrendhez a következő térkép tartozik:



Az alábbi algoritmus megvizsgálja, hogyan lehet eljutni Budapestről Nagykanizsára. Add meg, milyen sorrendben írja ki az algoritmus az egyes városokat!

A sor nevű adatszerkezetnek csak a végére lehet új elemet tenni (Sorba(I)), elvenni belőle pedig csak az elejéről lehet (I:=Sorból).

A Voltunk(I) eljárással feljegyezzük, hogy az I-edik városban már voltunk a bejárás során.

Bejárás:

Sorba(4): Voltunk(4)

Ciklus amíg a sor nem üres és nem voltunk Nagykanizsán

I:=Sorból

Ciklus J=1-től 9-ig

Ha a J-edik városban még nem voltunk és van közvetlen út az I-edikből a J-edikbe akkor Sorba(J): Voltunk(J): J-edik városnév kiírása

Ciklus vége

Ciklus vége

Eljárás vége.

Harmadik-ötödik osztályosok1. feladat (12 pont)

Az alábbi két függvényeljárás (nyissz és nyassz) rudakat darabol. A rudak hosszát a rúd tömbben tároljuk, amelynek a rudak számánál eggyel több eleme van. A tömb utolsó eleme az ún. strázsza, amelyben -1 van. A két függvényeljárás bemenő paramétereként azt kell megadni, hogy mekkora rúddarabot akarunk levágni.

```

VAR rúd: ARRAY [1..Maxrúd] OF INTEGER;
FUNCTION nyissz (h:INTEGER): INTEGER;
  VAR p: INTEGER;
BEGIN
  p := 1;
  WHILE (rúd[p+1] <> -1) AND (rúd[p] < h) DO p := p + 1;
  IF rúd[p] < h
    THEN nyissz := 0
    ELSE BEGIN nyissz := p; rúd[p] := rúd[p] - h END
END {nyissz};

FUNCTION nyassz (h:INTEGER): INTEGER;
  VAR p, q, m: INTEGER;
BEGIN
  p := 1; q := 0; m := MAXINT;
  WHILE (rúd[p] <> -1) DO
    BEGIN IF (rúd[p] - h >= 0) AND (rúd[p] - h < m)
      THEN BEGIN m := rúd[p] - h; q := p END;
      p := p + 1
    END;
  IF q <> 0 THEN rúd[q] := m;
  nyassz := q
END {nyassz};

```

- A. Hogyan alakul a tömb tartalma, és milyen értéket adnak vissza az egyes függvényeljárások, ha kezdetben a tömbben [3, 5, 2, 4, 7, -1] van, és a 3, 4, 2, 2 paraméterekkel hívjuk meg mind a nyissz, mind a nyassz eljárást, külön-külön, a megadott sorrendben?
- B. A [2, 1, -1] tömbhöz adj meg olyan rúdigénylést, amelyre a nyissz függvényeljárás 0-t fog valamikor visszaadni, a nyassz pedig soha. (Ha nincs ilyen, válaszodat akkor is indokold meg!)
- C. A [3, 2, -1] tömbhöz adj meg olyan rúdigénylést, amelyre a nyassz függvényeljárás fog valamikor 0-t visszaadni, a nyissz pedig soha. (Ha nincs ilyen, válaszodat akkor is indokold meg!)

2. feladat (10 pont)

Furcsa számítógépünk a következő formában várja tőlünk a programot. Először meg kell adni az alapismereteket. Másodszor föl kell sorolni azokat a következtetési szabályokat, amelyek alkalmazásával a problémát megoldhatónak gondoljuk.

Alapismeretek például:

1. anya(Szilágyi Erzsébet, Mátyás).
Ez azt jelenti, hogy Szilágyi Erzsébet anyja Mátyásnak.

2. apja(Hunyadi János, Mátyás).
Ez azt jelenti, hogy Hunyadi János apja Mátyásnak.

Szabályok például:

1. szülője(x,y) HA anyja(x,y) VAGY apja(x,y).
2. nagyszülője(x,y) HA szülője(x,z) ÉS szülője(z,y).
Ez magyarul a következőt jelenti: akkor nagyszülője x y-nak, ha van olyan z személy, akinek x a szülője és ő (azaz z) szülője y-nak.

A logikai kifejezéseket balról jobbra haladva értékeli ki a gép. A kiértékelést abbahagyja, ha a részeredmény alapján már eldönthető a teljes kifejezés értéke.

A: Milyen rokonsági kapcsolatot határoznak meg a következő szabályok?

- a. rokon1(x,y) HA szülője(z,x) ÉS szülője(z,y) ÉS $x < y$.
- b. rokon2(x,y) HA apja(x,z) ÉS szülője(z,y).
- c. rokon3(x,y) HA szülője(x,y) VAGY (szülője(x,z) ÉS rokon3(z,y)).

B: Írd meg a következő rokoni kapcsolatokat leíró szabályokat!

- a. anyaiNagyszülője(x,y) HA ... {azaz x anyai nagyszülője y-nak}
- b. szülőpár(x,y) HA ... {azaz akiknek közös gyermekük van}
- c. leszármazottja(x,y) HA ... {azaz x leszármazottja y-nak}

3. feladat (8 pont)

Egy palacsintasütő és egy palacsintaevő ember számára készítettünk egy-egy algoritmust. Ugy tervezzük, hogy a két ember ezeket az algoritmusokat egyszerre – azaz egymással párhuzamosan – hajtja végre. Egy közös tárolót (tányért) használnak, amelyen egyszerre csak egy palacsinta fér el. Egymással nem beszélhetnek, csupán egy-egy, a vasúttállomásokon alkalmazottakhoz hasonló semaforral jelezhetnek egymásnak. Ehhez a következő utasításokat használhatják fel:

jelezz(SZ) szabadra állítja az SZ semafort,
várj(SZ) várakozik, amíg az SZ semafor nem szabadot jelez, majd ismét tilosra állítja, és abbahagyja a várakozást.

Sütő:

Ciklus
Süss egy palacsintát!
Várj(ÜRES A TÁNYÉR)
Tedd a palacsintát
a tányérra!
Jelezz(EHETSZ)
Ciklus vége
Eljárás vége.

Evő:

Jelezz(ÜRES A TÁNYÉR)
Ciklus
*
Ciklus vége
Eljárás vége.

A * helyébe négy utasítást teszünk, különféle sorrendben. Add meg, hogy mely megoldások hibásak és miért! Ha több helyes megoldás is van, vizsgálj meg, hogy melyik mennyire hatékony!

A: Várj(EHETSZ)
Vedd fel a palacsintát!
Jelezz(ÜRES A TÁNYÉR)
Edd meg!

B: Vedd fel a palacsintát!
Várj(EHETSZ)
Jelezz(ÜRES A TÁNYÉR)
Edd meg!

C: Várj(EHETSZ)
 Jelezz(ÜRES A TÁNYÉR)
 Vedd fel a palacsintát!
 Edd meg!

D: Várj(EHETSZ)
 Vedd fel a palacsintát!
 Edd meg!
 Jelezz(ÜRES A TÁNYÉR)

4. feladat (15 pont)

A következő két rendezőeljáráshoz írd olyan állításokat, amelyek a megjelölt helyeken érvényesek, és a rendezendő vektorra vonatkozó minden lényeges információt tartalmaznak!

Rendezés:
 Ciklus $l=1$ -től $N-1$ -ig
 { A. állítás }
 $M:=l$
 Ciklus $J=l+1$ -től N -ig
 { B. állítás }
 Ha $A(M) > A(J)$
 akkor $M:=J$
 Ciklus vége
 Csere($A(l), A(M)$)
 Ciklus vége
 Eljárás vége.

Rendezés:
 Ciklus $l=2$ -től N -ig
 { C. állítás }
 $J:=l-1$
 Ciklus amíg $J > 0$ és $A(J) > A(J+1)$
 { D. állítás }
 Csere($A(J), A(J+1)$): $J:=J-1$
 Ciklus vége
 Ciklus vége
 Eljárás vége.

5. feladat (12 pont)

Adott az alábbi Pascal-nyelvű függvényeljárás:

```
FUNCTION hash(v: INTEGER): INTEGER;
  VAR h: INTEGER;
BEGIN
  h:=0;
  WHILE v <> 0 DO BEGIN h:=h+v MOD 10; v:=v DIV 10 END;
  hash:=h MOD Max
END {hash};
```

A hash függvényt a következő eljárás használja:

```
PROCEDURE insert(v: INTEGER);
  VAR place, n: INTEGER;
BEGIN
  place:=hash(v); n:=0;
  WHILE (n < Max) AND (memo[place] >= 0) DO
    BEGIN place:=(place+1) MOD Max; n:=n+1 END;
  IF memo[place] < 0 THEN memo[place]:=v
  END {insert};
```

A

VAR memo: ARRAY [0..Max-1] OF INTEGER

(globális) tömb minden elemének kezdetben -1 az értéke.

Mi lesz a memo tömb tartalma az alábbi insert-sorozatok adott sorrendű végrehajtása után, a Max állandó adott értéke mellett?

- A. CONST Max = 7;
insert(2); insert(26); insert(117); insert(8); insert(18)
- B. CONST Max = 11;
insert(2); insert(26); insert(117); insert(8);
insert(18); insert(14); insert(13)

6. feladat (12 pont)

Egy adatátviteli rendszerben két számítógép között soros vonalon, négybites csomagokban továbbítják az információt. Mivel az átviteli csatorna nem tökéletes, kb. minden 50. átküldött bit megsérül. Annak azonban elenyésző a valószínűsége, hogy egymáshoz ennél közelebb lévő bitek sérüljenek meg. Ezért minden négybites csomagot 7 biten kódolva küldenek át, így ha a 7 bitből csak egy sérül meg, akkor az eredetileg átküldött adat visszaállítható. Ha több bit sérül meg egyszerre, akkor az alábbi algoritmus nem működik helyesen. (A XOR bitenkénti KIZÁRÓ VAGY művelet.)

A következő programrészlet a vevőoldalon a dekódolás feladatát látja el. Bemenete a csat hételemű tömb, amely a csatornán érkezett és ugyanabba a csomagba tartozó biteket tartalmazza. Kimenatként az üzen négyelemű tömbben helyreállítja az eredeti üzenet 4 bitjét.

```
i := 0;
IF (csat[1] XOR csat[3] XOR csat[5] XOR csat[7]) = 1 THEN i := i + 1;
IF (csat[2] XOR csat[3] XOR csat[6] XOR csat[7]) = 1 THEN i := i + 2;
IF (csat[4] XOR csat[5] XOR csat[6] XOR csat[7]) = 1 THEN i := i + 4;
IF i > 0 THEN csat[i] := NOT(csat[i]);
üzen[1] := csat[3]; üzen[2] := csat[5];
üzen[3] := csat[6]; üzen[4] := csat[7]
```

A következő programrészlet az adóoldalon a kódolás feladatát látja el. Értelemszerűen az előző eljárás fordítottját végzi. Néhány szám azonban hiányzik belőle. Ezeket * -gal helyettesítettük.

```
csat[3] := üzen[1]; csat[5] := üzen[2];
csat[6] := üzen[3]; csat[7] := üzen[4];
csat[1] := csat[*] XOR csat[*] XOR csat[*];
csat[2] := csat[*] XOR csat[*] XOR csat[*];
csat[4] := csat[*] XOR csat[*] XOR csat[*]
```

Milyen értékeket vehet fel és mit fejez ki i hibátlan átvitel, ill. egyszeres hiba esetén? Add meg a * -gal helyettesített számokat a programban való előfordulásuk sorrendjében!

7. feladat (12 pont)

Pap Jancsi fogad Tasziló barátjával, hogy a padláson heverő lyukszalagok egyikén sem fordul elő a "LILLA" szó. De hogy biztos legyen a dolgában, programot ír a lyukszalagok átalakítására.

Az alábbi BASIC nyelvű programban az OLVAS(K\$) parancs az 1-es számú lyukszalag következő karakterét olvassa be K\$-ba, az IR(L\$) parancs pedig a 2-es számú lyukszalagra írja ki az L\$ tartalmát. A szalagok végét a "\$" jel jelzi. (Mivel Pap Jancsi még nem nagyon tud programozni, programja hibajelzéssel fog leállni a szalag végén.)

Pap Jancsi minden szalagot átmásol a programmal, majd a már átalakított szalagokat is újra meg újra átfuttatja, amíg csak változást tapasztal. Végül minden szalag eredeti helyett annak utolsó változatát rakja vissza a padlásra.

```

10 OLVAS(K$)
15 IF K$ <> "L" THEN IR(K$):GOTO 10 ELSE OLVAS(K$)
20 IF K$ <> "I" THEN IR("L"):GOTO 15 ELSE OLVAS(K$)
30 IF K$ <> "L" THEN IR("LI"):GOTO 15 ELSE OLVAS(K$)
40 IF K$ <> "L" THEN IR("LIL"):GOTO 15 ELSE OLVAS(K$)
50 IF K$ <> "A" THEN IR("LILL"):GOTO 15 ELSE GOTO 10
60 END

```

Tasziló, aki megneszelte a turpisságot, már előre dörzsöli a kezét. Jogosan, mert a program valóban rossz.

- A: Tasziló, hogy nyilvánvalóvá tegye diadalát, saját készítésű lyukszalagot dug el Pap Jancsi padlásán. Mi legyen ezen a szalagon, hogy a program biztosan felsüljön? (Azaz benne hagyjon legalább egy LILLAt.)
- B: Pap Jancsi közben maga is rájön a hibára, és kijavítja a programot. Két sorban végez apró módosítást, a többi soron nem változtat, és új sort sem szúr be. Add meg ezt a két sort kijavítva!

8. feladat (13 pont)

A Folttsztító Turbo-Pascal eljárás a grafikus képernyőn lévő és a háttértől elütő színű alakzat pontjait festi át háttérszínűre, bizonyos rendszer szerint. A felhasznált grafikus utasítások és függvények:

GetPixel(x,y): az (x,y) koordinátájú pont színét adja vissza,
 GetBkColor: a háttér színét adja vissza,
 PutPixel(x,y,szín): az (x,y) pontot szín színűre festi.

Megjegyzés: az origó a bal felső sarokban van, x vízszintes, y függőleges elmozdulást jelent.

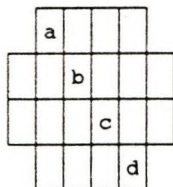
```

PROCEDURE Folttsztító (x, y: INTEGER);
  VAR x1, y1, d, n: INTEGER;
BEGIN
  x1 := x; y1 := y; d := 1; n := 0;
  REPEAT
    CASE d OF
      1: BEGIN x1 := x+1; y1 := y END;
      2: BEGIN x1 := x; y1 := y+1 END;
      3: BEGIN x1 := x-1; y1 := y END;
      4: BEGIN x1 := x; y1 := y-1 END
    END {CASE};
    IF (GetPixel(x1, y1) <> GetBkColor)
      THEN BEGIN PutPixel(x, y, GetBkColor); n := 0;
              x := x1; y := y1; d := (d+2) MOD 4 + 1
            END
      ELSE BEGIN d := d MOD 4 + 1; n := n + 1 END
    UNTIL n = 4
  END {Folttsztító};

```

- A. Mi a d változó szerepe, és mit jelentenek egyes értékei?
- B. A képernyőn a következő, a háttértől elütő színű alakzat helyezkedik el. (Egy-egy mező egy-egy képernyőpontot jelent.) Sorszámozd be a program által "ki-

tisztított" pontokat a tisztítás sorrendjében, az a, b, c és d betűkkel megjelölt, négy különböző kiindulási mező esetén! (Mind a négy esetre rajzolj egy-egy ábrát!)



1991. Második forduló

Első-második osztályosok: 1. Hetedhét

A 'Hét szűk esztendő' albizottság hétepcséses titokként őrzi tervezetét, amelyet a bennfentesek 'Hetet egy csapásra' törvénynek becéznek. Segítségre van szükségük, ezért fordulnak hozzád.

Csak akkor olvasd tovább, ha tudsz titkot tartani!

A tervezet egyik sarkalatos pontja szerint gazdasági gondjainkat az ősmagyar hetes számrendszer bevezetésével meg lehet oldani, ugyanis a hetes számnak köztudottan mágikus hatása van. A többség a helyi érték nélküli számábrázolás mellett tör lándzsát, mondván, hogy akkor a számok hosszabbak, és a fizetések többnek látszanak. (A hétpróbás ellenzék hetet-havat összehord, amikor azt állítja, hogy az árak is magasabbnak tűnnek majd fel.)

A javaslat szerint az új számrendszer számjegyeit, az ún. szűkszámjegyeket a hét honfoglaló törzsfőről nevezik el, és jelölésükre e nevek egyik jellemző betűjét használják. (Zárójelben megadjuk a számjegyeket jelölő betűket, és - 7 hatványaként - az elavult tízes számrendszerbeli megfelelőiket.)

Álmos	(A = 1)	Előd	(E = 7)	Ord	(O = 7 ²)
Kont	(K = 7 ³)	Tas	(T = 7 ⁴)	Huba	(H = 7 ⁵)
Töhötöm	(M = 7 ⁶)				

(Jelenleg arról folyik még vita, hogy a nevek írásánál és ejtésénél a Gesta Hungarorumot avagy Anonymust tekintsek-e hiteles forrásnak.)

Az első ötven szűkszámot az alábbi táblázat mutatja.

	+1	+2	+3	+4	+5	+6	+7
0	A	AA	AAA	AAAA	AAE	AE	E
7	EA	EAA	AAA	EAAA	EAAE	EAE	EE
14	EEA	EEAA	EEAAA	EEAAA	EEAAE	EEAE	EEE
21	EEEE	EEEEA	EEEEAA	EEEEAAE	EEEEAAE	EEEEAE	EEEE
28	EEEEOA	EEEEOAA	EEEEOAAA		EEEEOAAE	EEEEOAAE	EEEE
35	EEEOA	EEEOAA	EEEOAAA	EEEOAAE	EEEOAAE	EEEOAE	EEEO
42	EOA	EOAA	EOAAA	AAAO	AAO	AO	O
49	OA						

A szűkszámok képzésének szabályai:

Szükszámjegyet legfeljebb három nála kisebb, de egyforma számszámjegy előzhet meg. A kisebb számjegyeknek közvetlenül a nagyobb számjegy előtt kell állniuk, és ilyenkor a kisebb jegyek értékét le kell vonni a nagyobb számjegy értékéből. Ettől eltekintve a számszámok számjegyeit balról jobbra csökkenő sorrendben kell felírni, de úgy, hogy bármely számszámjegy után közvetlenül legfeljebb két vele azonos számszámjegy állhat. Ilyenkor a számjegyek értékét össze kell adni. (Ha e szabályok alkalmazásával többféle alak is lehetséges, akkor az a helyes, amely a lehető legkevesebb számszámjegyből áll. Pl. EOAE helyett AO a helyes alak.)

Egyetlen gondja van még a héttagú bizottságnak: abban a rövid időszakban, amíg az emberek megszokják a számszámokat, minden számszámot át kell alakítani tízes számrendszerbeli számmá. Ehhez kell a te segítséged!

Írj olyan programot, amely tetszőleges számszámot átalakít tízes számrendszerbeli számmá!

A programnak csak szintaktikailag helyes számokat adunk, és a számok helyességét nem kell ellenőriznie. Az átalakítandó (garantáltan helyes) számszámokat a SZUK.BE szöveges állományból kell beolvasni; minden számszámot új sorba írunk, az első számszám az első sorban van. A számszámok előtt és mögött tetszőleges számú szóköz lehet. A számszám tízes számrendszerbeli megfelelőjét a képernyőre és a SZUK.KI nevű állományba kell kiírni. Minden számot külön sorba kell rakni; az első szám az első sorban legyen.

Példaként megadunk egy minta SZUK.BE állományt. Ezt tetszés szerint módosíthatod, kiegészítheted. A programodat azonban más adatokkal is ki fogjuk próbálni!

SZUK.BE	SZUK.KI
EAA	9
EEEO	28
HHAH	50420
MHT	136857
H	16807
HHAH	50420

Megjegyzés:

A 16-bites számbábrázolás korlátai miatt a C64 és C+4 gépeken helyesnek fogadjuk el azokat a programokat is, amelyek csak a H-nál nem nagyobb számszámokat kezelik jól.

Értékelés (40 pont):

A. Beolvasás	3 pont
B. Kiírás	3 pont
C. Összes számszámjegy felismerése (PC-n: Hosszú egészek kezelése is)	7 pont
D. Előtagok kivonása	7 pont
E. Utótagok hozzáadása	7 pont
F. Szóközök szűrése	3 pont
G. Programszerkezet, stílus	10 pont

Beadandó:

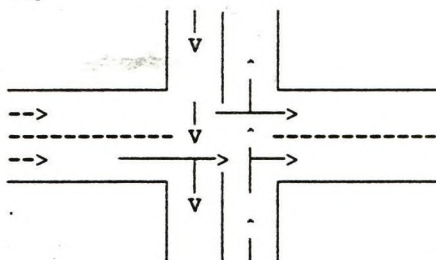
A program szövege lemezen és kinyomtatva; minden, a program futtatásához szükséges segédprogram, szubrutin lemezen; minden olyan dokumentáció, amely a megoldás megértését és értékelését segíti (programvázlat, adatok és adatszerkezetek specifikációja stb.).

Első–második osztályosok: 2. Útkereszteződés

Készíts programot, amely egy útkereszteződés forgalmát szimulálja! Az autók a két útszakasz elejére véletlenszerűen érkeznek, az úton végigmennek, majd az útszakasz végén eltűnnek. Az ábrán függőlegesen rajzolt út kétirányú, a felfelé menő sávból lehet jobbra is kanyarodni. A vízszintes út kétsávos, a jobb oldali sávból jobbra, a bal oldaliból pedig balra is lehet kanyarodni. A kereszteződés forgalmát jelzőlámpák irányítják.

Megoldandó részfeladatok:

1. A kereszteződés és az éppen látható autók kirajzolása.
2. A jelzőlámpák pillanatnyi állapotának megjelenítése.
3. Az utakon mozgó vagy várakozó autók számának kijelzése.
4. Az autók a két útszakaszra véletlenszerűen érkeznek (ez legyen paraméterezhető), és az úton végigmennek (addig, ameddig a képernyőn ábrázoljuk az utat).
5. Az autók a kereszteződésben vagy egyenesen haladnak át, vagy a megfelelő irányba fordulnak.
6. Az autók a közlekedési lámpák miatt, ill. az előttük megálló autók miatt az úton megállhatnak. Amikor lehetséges, tovább kell indulniuk.
7. A lámpaváltások idejét a felhasználó adja meg, amelyet a programnak a forgalomtól függően módosítania kell.
8. Az autók vagy állnak, vagy azonos sebességgel haladnak, egymást nem előzhetik meg.



Értékelés (100 pont):

A.	Ábrázolás	14 pont
B.	Autók mozgása	26 pont
C.	Lámpakezelés	20 pont
D.	Autók megállása, elindulása	15 pont
E.	Programszerkezet	25 pont

Beadandó:

A program szövege lemezen és kinyomtatva; minden, a program futtatásához szükséges segédprogram, szubrutin lemezen, minden olyan dokumentáció, amely a megoldás értékelését segíti (programvázlat, adatok, adatszerkezetek specifikációja stb).

Harmadik–ötödik osztályosok: Szó ami szó

Napjainkban egyre fontosabb a számítógépek nem műszaki területeken való alkalmazása. Ez általában adatbázisokon, szövegeken való műveleteket jelent. Fontos feladattípus az, amikor nem egyszerűen csak szavakat, karaktersorozatokat kere-

sünk egy szövegben, hanem bizonyos szavak, szótöredékek, szócsoportok együttes előfordulását (szakszóval ezt nevezik információ-visszakérésnek). Alkothatunk például logikai kifejezést a keresni kívánt szavakból: "Keressük azt a szövegrészt, amelyben előfordul az egér szó, továbbá a kutya vagy a macska szavak közül legalább az egyik." Ennek formális leírására az "egér & (kutya + macska)" logikai kifejezés alkalmas, amelyben már a későbbiekben bevezetendő jelöléseket alkalmaztunk.

Írj programot, amely ilyen keresést valósít meg egy szövegben! A program tegyen eleget a következő kikötéseknek:

A szöveg

Az átvizsgálandó szöveg egy fájlban található, amelynek a neve legyen SZOVEG.INP. (Legjobb, ha ezt a program magától tudja, és nem kérdez rá a névre.)

A szöveg bekezdésekből áll. Két bekezdést egy üres sor választ el egymástól, az utolsó bekezdés a fájl végéig tart. Egy bekezdés több, tetszőleges hosszúságú, nem üres sorból állhat, de teljes hossza nem haladhatja meg a 250 karaktert.

A bekezdés sorai szavakból állnak, amelyeket írásjelek (szóköz, ''', ''', ''', '?' és '!') tetszőleges kombinációja választ el egymástól. Egy szó az ékezet nélküli ábécé kis- és nagybetűit, ill. számjegyeket tartalmazhat.

A program feltételezheti, hogy a bemeneti szöveg megfelel e szabályoknak, és ezt ellenőriznie sem kell.

A keresendő minta

A keresendő minta logikai kifejezés formájában adható meg. Ez mintaszavakból, az ÉS, VAGY, TAGADÁS műveletekből, valamint zárójelekből állhat. A műveletek precedenciája a szokásos (TAGADÁS > ÉS > VAGY); ezt azonban zárójellezéssel befolyásolni lehet.

A zárójelek a szokásos (és) karakterek. A műveleteket egy-egy karakter jelöli. Ezek a következők:

VAGY + ÉS & TAGADÁS !

A mintaszavak ékezet nélküli kis- és nagybetűket, számjegyeket, valamint ún. joker karaktereket tartalmazhatnak. Két mintaszó között a műveleti és zárójeleken kívül jelentés nélküli szóköz karakterek is állhatnak. (Ezek egyik szónak sem részei.)

A joker karaktert * jelöli. Ez egy szón belül bármilyen karaktersorozatra illeszkedik, beleértve az üres karaktersorozatot is. Joker a mintaszóban tetszőleges helyen előfordulhat, egy mintaszóban akár több helyen is; ilyenkor a különböző helyeken álló jokerek különböző karaktersorozatokat jelenthetnek. A mintaszó akkor illeszkedik egy szóra, ha a mintaszóban levő joker karakterek helyére a szabályoknak megfelelő karaktersorozatot illesztve a szó előállítható.

Az illeszkedés vizsgálatokor a kis- és nagybetűk között nem szabad különbséget tenni.

Példaként megadjuk, hogy a következő mintaszavak a hat, Alma, sok, HATALMAS, Heves, hasal szavak közül melyekre illeszkednek.

Minta	Illeszkedő szó
Hat	hat
hat*	hat, HATALMAS
airna	Alma, HATALMAS

S
H*s

sok, HATALMAS, Heves, hasal
HATALMAS, Heves

A program adjon lehetőséget arra, hogy a mintát akár a billentyűzetről, akár szöveges állományból beadhassuk. Ez utóbbi esetben a fájl neve MINTA.INP, és a minta egyetlen sorban helyezkedik el.

A keresés

A programnak azokat a bekezdéseket kell a bemeneti szövegből kiválasztania, amelyek megfelelnek a mintaként megadott logikai kifejezésnek. Egy bekezdés akkor felel meg a kifejezésnek, ha a kifejezésben lévő mintaszavak logikai értékekkel való helyettesítése után a kifejezés logikai IGAZ értéket ad. A mintaszavak helyettesítési értéke IGAZ, ha a bekezdésben található a mintaszóra illeszkedő szó, és HAMIS, ha nem található ilyen. A különböző bekezdéseket a programnak egymástól függetlenül kell megvizsgálnia.

Eredmény

A program írja ki a szövegből a mintának megfelelő bekezdéseket a képernyőre és az EREDM.OUT nevű fájlba. A program a szöveg feldolgozásának végén üzenetben tudassa a felhasználóval a talált bekezdések számát.

Példaként találsz a lemezen egy egyszerű szöveget, amelyet tesztelési célra használhatsz, és tetszés szerint bővíthetsz, módosíthatsz. A programodat azonban bonyolultabb szövegekkel és különféle mintákkal fogjuk kipróbálni.

Értékelés (140 pont):

A. Szövegbeolvasás	5 pont
B. Mintabeolvasás	5 pont
C. Kiírás	8 pont
D. Bekezdéskezelés	7 pont
E. Szóillesztés	25 pont
F. Kifejezéskiértékelés	60 pont
G. Programszerkezet, stílus	30 pont

Ha nem tudtad befejezni a feladatot, akkor feltétlenül írd le, hogy szerinted mely részfeladatokat sikerült megoldanod!

Beadandó

A program szövege lemezen és kinyomtatva; minden, a program futtatásához szükséges segédprogram, szubrutin lemeze; minden olyan dokumentáció, amely a megoldás megértését és értékelését segíti (programvázlat, adatok és adatszerkezetek specifikációja stb.).

1992. Első forduló

Első-második osztályosok

1. feladat (10 pont)

D.E. Knuth: A számítógép-programozás művészete című könyvében J. Stein alábbi algoritmusát közli:

"A program bemenő adatai az U és V természetes számok.

- B1. Legyen $k < 0$, majd ismételten $k <- k+1$, $U <- U/2$, $V <- V/2$, egyszer vagy többször, esetleg egyszer sem, mindaddig, amíg már U és V nem mindketten párosak.
- B2. Ha U páratlan, legyen $T <- -V$, és menjünk B4-re. Egyébként pedig legyen $T <- U$!
- B3. Legyen $T <- T/2$.
- B4. Ha T páros, menjünk vissza B3-ra.
- B5. Ha $T > 0$, legyen $U <- T$, egyébként legyen $V <- -T$.
- B6. Legyen $T <- U-V$! Ha $T < 0$, menjünk vissza B3-ra. Egyébként az algoritmus végét ér, az output $U^2 \cdot K$."

- A. Mi lesz a program eredménye a (15,20), (24,28), (54,60) számpárokra?
 B. Milyen feladatot old meg a program?

2. feladat (15 pont)

Az alábbi programrészlet az A\$ változó alapján állít elő egy szöveget a B\$-ban. A\$ csak nyomtatható karaktereket tartalmazhat! A felhasznált függvényeljárások jelentése:

RIGHT\$(T\$, i):	a T\$ szöveg jobb szélső i db karaktere,
LEFT\$(T\$, i):	a T\$ szöveg bal szélső i db karaktere,
MID\$(T\$, k, i):	a T\$ szöveg középső i db karaktere a k.-tól kezdve,
LEN(T\$):	a T\$ szöveg karaktereinek száma,
CHR\$(k):	a k kódú karakter.

```

1000 S = 1: B$ = LEFT$(A$, 1)
1010 IF B$ = A$ THEN 1200
1020 FOR I = 2 TO LEN(A$)
1030 F RIGHT$(B$, 1) = MID$(A$, I, 1) THEN S = S + 1: GOTO 1090
1040 F S > 2 THEN B$ = B$ + CHR$(0) + CHR$(S)
1050 F S = 2 THEN B$ = B$ + RIGHT$(B$, 1)
1060 B$ = B$ + MID$(A$, I, 1): S = 1
1090 NEXT
1200 ...

```

- A. Mi lesz B\$-ban az 1200-as sorban, ha A\$ tartalma kezdetben
- "Ipafai fapipa"
 - "Ézüsttel befuttatott."
 - "13333333-szor ismételd meg."
 - "Gyakorisága 888888"
- B. Fogalmazd meg tömören, mit csinál a fenti program!
 C. Milyen esetekben nem lehet előállítani B\$-ből A\$ értékét?
 D. Egészítsd ki a programot két új sorral (1100 és 1110) úgy, hogy B\$ből A\$ mindig előállítható legyen!

3. feladat (13 pont)

A következő programrészlet bemenete az $A(N)$ vektor, értékei 0 és $M-1$ közötti egész számok.

- A. Mit tartalmaz a megoldásban a B vektor a (*) ponton?
 B. Mit tartalmaz a megoldásban a B vektor a (**) ponton?
 C. Mit tartalmaz a megoldásban a B és a C vektor a (***) ponton?

Ciklus I := 0-tól (M-1)-ig

B(I) := 0

Ciklus vége

Ciklus I := 1-től N-ig

B(A(I)) := B(A(I)) + 1

Ciklus vége

(*)

Ciklus I := 1-től (M-1)-ig

B(I) := B(I) + B(I-1)

Ciklus vége

(**)

Ciklus I := 1-től N-ig

C(B(A(I))) := A(I); B(A(I)) := B(A(I)) - 1

Ciklus vége

(***)

4. feladat (18 pont)

Egy kétsávos, egyirányú út forgalmát szimuláló programot készítettünk. A sávok forgalmát az A tömb írja le:

```

      _____
A (N, 2)  ->  - - - - -
      ->  _____
  
```

$A(I,1) = -1$ azt jelenti, hogy a felső sáv I. helyén nincs autó; $A(I,1) \neq 0$ a felső sáv I. helyén található autó sebességét adja meg (azaz egy időegység alatt ennyit lépne előre). $A(I,2)$ ugyanezt jelenti az alsó sávra.

Add meg, milyen forgalmi szituációkat szimulálnak az F1, ..., F5 eljárások közül a megírtak, ill. írd meg a szövegesen megadottakat! (A megoldásban S a maximális sebességet jelenti.)

Ciklus I := N-től 1-ig -1-esével

Ciklus J := 1-től 2-ig

F1(I, A(I,J), J, K) : F1(I, A(I,J), 3-J, L)

Ha $K = A(I,J)$ és $A(I,J) < S$ akkor F2(I, J)

különben ha $K = A(I,J)$ akkor F3(I, J)

különben ha $L < A(I,J)$ akkor F4(I, J, K)

különben F5(I, J, L)

Ciklus vége

Ciklus vége

F1(I, T, J, K):

Lehet-e a J. sávban az I. helyről T lépést előrelépni? K legyen az a lépésszám, amennyit lehet (maximum T, minimum 0); $I+K = N+1$, ha az út végére ért az autó.

Eljárás vége.

F2(I, J):

Előrelépés a sebességnek megfelelő távolságra, majd gyorsítás 1 egységgel.

Eljárás vége.

F3(I, J):

Előrelépés a sebességnek megfelelő távolságra.

Eljárás vége.

F4(I, J, K):

$$A(I+K, J) := K-1 : A(I, J) := -1$$

Eljárás vége.

F5(I, J, L):

$$A(I+L, 3-J) := A(I, J) : A(I, J) := -1$$

Eljárás vége.

5. feladat (8 pont)

Mit, milyen helyzetben és méretben rajzolnak a következő, LOGO nyelvű programok? Az egyes parancsok jelentése:

LEFT f, RIGHT f:	balra-, ill. jobbrafordulás helyben f fokkal;
FORWARD h:	előrelépés h egységgel, közben rajzol, ha kell
PENUP, PENDOWN:	toil felemelése, ill. letevése a papírra
REPEAT db [utasítások]:	utasítások megismétlése db-szer.

A. LEFT 30

REPEAT 3

[REPEAT 2 [FORWARD x RIGHT 60 FORWARD x RIGHT 120] RIGHT 120]

B. REPEAT 4

[REPEAT 360 [FORWARD 1 RIGHT 1]

PENUP RIGHT 90 FORWARD 10 LEFT 90 PENDOWN]

6. feladat (13 pont)

Egy kurzormozgató program a kurzort a J, B, L, F billentyűkkel mozgatja. Mi a programban felhasznált többi billentyű hatása?

A 'Rajzol' eljárás egy szakasz pontjait váltja az ellenkezőjére a képen (sötétet világosra, ill. fordítva).

Kurzormozgatás

$$M := 1 : X := 100 : Y := 100 : RX := X : RY := Y : \text{Rajzol}(X, Y, X, Y)$$

Ciklus

Be: B

Elágazás

$$B = 'F' \text{ esetén } XX := X : YY := Y - 1 : \text{Változás}$$

$$B = 'L' \text{ esetén } XX := X : YY := Y + 1 : \text{Változás}$$

$$B = 'J' \text{ esetén } XX := X + 1 : YY := Y : \text{Változás}$$

$$B = 'B' \text{ esetén } XX := X - 1 : YY := Y : \text{Változás}$$

$$B = '1' \text{ esetén } M := 1 : RX := X : RY := Y : \text{Rajzol}(X, Y, X, Y)$$

$$B = '2' \text{ esetén } M := 2 : RX := X : RY := Y : \text{Rajzol}(X, Y, X, Y)$$

Elágazás vége

Ciklus vége

Eljárás vége.

Változás:

Ha $M = 1$ akkor
különben
$$\text{Rajzol}(RX, RY, X, Y) : \text{Rajzol}(RX, RY, XX, YY)$$

$$\text{Rajzol}(RX, RY, RX, Y) : \text{Rajzol}(RX, RY, RX, YY)$$

$$\text{Rajzol}(RX, RY, X, RY) : \text{Rajzol}(RX, RY, XX, RY)$$

$$\text{Rajzol}(RX, Y, X, Y) : \text{Rajzol}(RX, YY, XX, YY)$$

$$\text{Rajzol}(X, RY, X, Y) : \text{Rajzol}(XX, RY, XX, YY)$$

Elágazás vége

 $X := XX : Y := YY$

Eljárás vége.

7. feladat (10 pont)

Egy színes monitor a képen megjelenő színeket az RGB színmodell segítségével állítja elő (R (red) = piros, G (green) = zöld, B (blue) = kék). A megjelenő szín fehér, ha mindhárom alapszín benne van; fekete, ha egyik sincs benne; alapszín, ha a három közül csak az egyik fordul elő benne; és keverékszín, ha kettő. A színeket tehát egy három elemű logikai vektorral ábrázoljuk, azoknál az alapszíneknél igaz értékekkel, amelyek az adott színben előfordulnak. Milyen műveleteket végeznek a következő eljárások?

E1(S1, S2, S3):

Ciklus S := red-től blue-ig

 $S3(S) := S1(S) \text{ OR } S2(S)$

Ciklus vége

Eljárás vége.

E2(S1, S2):

Ciklus S := red-től blue-ig

 $S2(S) := \text{NOT } S1(S)$

Ciklus vége

Eljárás vége.

E3(S1, SZ, I):

 $I := 1$

Ciklus S := red-től blue-ig

Ciklus J := 1-től 3-ig

 $SZ(J, S) := \text{hamis}$

Ciklus vége

 $SZ(I, S) := S1(S)$ Ha $S1(S)$ akkor $I := I + 1$

Ciklus vége

Eljárás vége.

8. feladat (13 pont)

Mit adnak eredményül a következő függvényeljárások (ahol L, L1, L2 karaktersorozatok, E pedig egy karakter):

A.

Ismerjfel(L1, L2):

Ha üres(L1) akkor eredmény := L2

különben eredmény := egymásután(első(L1), Ismerjfel(elsőtúniak(L1), L2))

Függvény vége.

B.

Találjki(E, L):

Ha üres(L) akkor eredmény := L

különben Ha E = első(L) akkor eredmény := elsőtúniak(L)

különben eredmény := egymásután(első(L),

Találjki(E, elsőtúniak(L)))

Függvény vége.

C.

Mi az eredménye az

Ismerjfel("NEMES", egymásután("+", "TIHAMÉR"))

függvényeljárás-hívásnak?

D.

Mi az eredménye a Találjki("E", "NEMES") függvényeljárás-hívásnak?

Harmadik-ötödik osztályosok1. feladat (18 pont)

A következő függvényeljárás az A tömb elemeiből számít ki egy értéket (a tömböt 1-től indexeljük):

F(P):

Ha A(P) ≠ 0 akkor F := A(P)

különben I := F(2*P); J := F(2*P+1)

Ha I = -1 vagy J = -1 akkor F := -1

különben Ha A(P) = -1 akkor F := I+J

különben Ha A(P) = -2 akkor F := I-J

különben Ha A(P) = -3 akkor F := I*J

különben Ha A(P) = -4 akkor Ha J = 0 akkor F := -1

különben F := I/J

különben F := -1

Elágazások vége

Eljárás vége.

A. Mi az A tömbben lévő számok pontos szerepe, jelentése?

B. Mi lesz F(1) értéke, ha az A tömb tartalma [-2, 7, 5]?

C. Mi lesz F(1) értéke, ha az A tömb tartalma [-1, -3, -4, 2, 2, 6, 2]?

D. Milyen esetekben lesz az F(1) függvényeljárás-hívásnak -1 az eredménye?

2. feladat (15 pont)

Az alábbi programrészlet az A\$ változó alapján állít elő egy szöveget a B\$-ban. A\$ csak nyomtatható karaktereket tartalmazhat! A felhasznált függvényeljárások jelentése:

RIGHT\$(T\$, i): a T\$ szöveg jobb szélső i db karaktere,
 LEFT\$(T\$, i): a T\$ szöveg bal szélső i db karaktere,
 MID\$(T\$, k, i): a T\$ szöveg középső i db karaktere a k.-tól kezdve,
 LEN(T\$): a T\$ szöveg karaktereinek száma,
 CHR\$(k): a k kódú karakter.

```

1000 S = 1: B$ = LEFT$(A$, 1)
1010 IF B$ = A$ THEN 1200
1020 FOR I = 2 TO LEN(A$)
1030 F RIGHT$(B$, 1) = MID$(A$, I, 1) THEN S = S + 1: GOTO 1090
1040 F S > 2 THEN B$ = B$ + CHR$(0) + CHR$(S)
1050 F S = 2 THEN B$ = B$ + RIGHT$(B$, 1)
1060 $ = B$ + MID$(A$, I, 1): S = 1
1090 NEXT
1200 ...

```

- A. Mi lesz B\$-ban az 1200-as sorban, ha A\$ tartalma kezdetben
- "Ipaiai fapipa"
 - "Ezüsttel befuttatott."
 - "13333333-szor ismételd meg."
 - "Gyakorisága 888888"
- B. Fogalmazd meg tömören, mit csinál a fenti program!
- C. Milyen esetekben nem lehet előállítani B\$-ből A\$ értékét?
- D. Egészítsd ki a programot két új sorral (1100 és 1110) úgy, hogy B\$ből A\$ mindig előállítható legyen!

3. feladat (13 pont)

A következő programrészlet bemenete az A(N) vektor, értékei 0 és M-1 között egész számok.

- Mit tartalmaz a megoldásban a B vektor a (*) ponton?
- Mit tartalmaz a megoldásban a B vektor a (**) ponton?
- Mit tartalmaz a megoldásban a B és a C vektor a (***) ponton?

Ciklus l := 0-tól (M-1)-ig

B(l) := 0

Ciklus vége

Ciklus l := 1-től N-ig

B(A(l)) := B(A(l)) + 1

Ciklus vége

(*)

Ciklus l := 1-től (M-1)-ig

B(l) := B(l) + B(l-1)

Ciklus vége

(**)

Ciklus l := 1-től N-ig

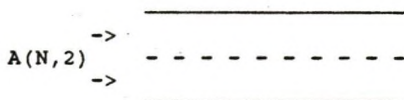
C(B(A(l))) := A(l): B(A(l)) := B(A(l)) - 1

Ciklus vége

(***)

4. feladat (18 pont)

Egy kétsávos, egyirányú út forgalmát szimuláló programot készítettünk. A sávok forgalmát az A tömb írja le:



$A(I, 1) = -1$ azt jelenti, hogy a felső sáv I . helyén nincs autó; $A(I, 1) \neq 0$ a felső sáv I . helyén található autó sebességét adja meg (azaz egy időegység alatt ennyit léphet előre). $A(I, 2)$ ugyanezt jelenti az alsó sávra.

Add meg, milyen forgalmi szituációkat szimulálnak az $F1, \dots, F5$ eljárások közül a megírtak, ill. írd meg a szövegesen megadottakat! (A megoldásban S a maximális sebességet jelenti.)

Ciklus $I := N$ -től 1 -ig -1 -esével

Ciklus $J := 1$ -től 2 -ig

$F1(I, A(I, J), J, K) : F1(I, A(I, J), 3-J, L)$

Ha $K = A(I, J)$ és $A(I, J) < S$ akkor $F2(I, J)$

különben ha $K = A(I, J)$ akkor $F3(I, J)$

különben ha $L < A(I, J)$ akkor $F4(I, J, K)$

különben $F5(I, J, L)$

Ciklus vége

Ciklus vége

$F1(I, T, J, K)$:

Lehet-e a J . sávban az I . helyről T lépést előrelépni? K legyen az a lépésszám, amennyit lehet (maximum T , minimum 0); $I+K = N+1$, ha az út végére ért az autó.

Eljárás vége.

$F2(I, J)$:

Előrelépés a sebességnek megfelelő távolságra, majd gyorsítás 1 egységgel.

Eljárás vége.

$F3(I, J)$:

Előrelépés a sebességnek megfelelő távolságra.

Eljárás vége.

$F4(I, J, K)$:

$A(I+K, J) := K-1 : A(I, J) := -1$

Eljárás vége.

$F5(I, J, L)$:

$A(I+L, 3-J) := A(I, J) : A(I, J) := -1$

Eljárás vége.

5. feladat (14 pont):

A T tömbben tároljuk egy terület $N \times M$ pontjának tengerszint feletti magasságát. A pontok vetülete egymástól S távolságra van egy négyzetrácson. Meg akarjuk rajzoltatni az SZ magasságú szintvonalat. Csak olyan SZ értéket választunk, hogy egyetlen rácspont se essen szintvonalra. Rajzolásra a következő programrészletet használjuk:

Ciklus I := 1-től (N-1)-ig
 Ciklus J := 1-től (M-1)-ig
 Ha Szintvonal(I, J, I+1, J) akkor Szakaszcrajz(I'S-S/2, J'S-S,
 I'S-S/2, J'S-S/2)
 Ha Szintvonal(I, J, I, J+1) akkor Szakaszcrajz(I'S-S, J'S-S/2,
 I'S-S/2, J'S-S/2)
 Ha Szintvonal(I+1, J, I+1, J+1) akkor Szakaszcrajz(I'S, J'S-S/2,
 I'S-S/2, J'S-S/2)
 Ha Szintvonal(I, J+1, I+1, J+1) akkor Szakaszcrajz(I'S-S/2, J'S,
 I'S-S/2, J'S-S/2)
 Ciklus vége
 Ciklus vége.

Szintvonal(I, J, K, L):

Szintvonal := (T(I, J) < SZ ÉS T(K, L) > SZ) VAGY (T(I, J) > SZ ÉS T(K, L) < SZ)
 Eljárás vége.

Ez a megoldás pontnégyesekre bontja a síkot. Ha a pontnégyes egy pontpárja között átmegy a Szintvonal, akkor a Szakaszcrajz a pontpár által meghatározott szakasz felezőpontját összeköti a pontnégyes által meghatározott négyzet középpontjával. Pl. az alábbi ábrát kapjuk, ha feltesszük, hogy két szomszédos pontpár között halad át a szintvonal (a-val SZ alatti, f-fel SZ feletti pontot jelölünk):



- A) Azt akarjuk, hogy a szintvonal egy pontnégyesen belül a pontpárok közötti szakaszokat – ha áthalad rajtuk – ne felezzé, hanem úgy messe, hogy a két rész hosszának aránya azonos legyen a két részre eső magasságkülönbségek arányával! Feltétve, hogy létezik, add meg az (I, J) és (I, J+1) közötti metszéspont koordinátáit!
- B) Sorold fel, hogy egy pontnégyes pontjainak magasságától függően hányféleképpen és hány helyen metszheti az A) pont szerinti SZ magasságú szintvonal a pontnégyes oldalait!

6. feladat (16 pont)

Egy agyafúrt bankrabló lefotózta a széf biztonsági programját. Balszerencséjére a programrész eleje és vége lemaradt!

- A. Milyen értékeket vehet fel a B változó a 12310-es sorban?
 B. Hogyan változtatja a ciklusmag az A és a B értékét?
 C. Legfeljebb hány kódot kell kipróbálnia a betörőnek, hogy biztosan kinyithassa a széfet, és melyek lehetnek ezek?


```

--
12300 B = INT(ABS(X))
12310 B = B-INT(B/3)*3+1
12320 INPUT "Mi a titkos kód", A
12325 IF A < 100 OR A > 999 THEN GOTO 07
12330 A = INT(A-33)*2
12340 FOR I = 1 TO 2
12345 C = INT(A/10)
12350 B = B+A-C*10
12360 A = C
12370 NEXT
12380 IF B <> A OR B > 3 THEN GOTO 12300
12390 PRINT "Máris nyitom!"
--

```

7. feladat (13 pont)

Az alábbi rekurzív függvényeljárás bemenő paraméterei az X, M, A és B változók. X és M csak pozitív értékek lehetnek.

- Milyen értéket számít ki a függvény?
- Mi M szerepe?
- Mi A és B szerepe?
- Tetszőleges pozitív X és M esetén adj meg olyan A és B kezdőértékpárt, amely biztosan jól!

```

Function Rekurzív(X, M, A, B: Real): Real;
  Var C: Real;
Begin
  C := (A+B)/2;
  If C-A < M Then Rekurzív := C
  Else If C*C > X Then Rekurzív := Rekurzív(X, M, A, C)
  Else Rekurzív := Rekurzív(X, M, C, B)
End {Rekurzív};

```

8. feladat (13 pont)

A lista mint adatszerkezet rekurzív definíciója a következő:

- egyetlen elemet sem tartalmaz, azaz üres;
- egy nemüres elemet és egy azt követő listát tartalmaz.

A lista-adattípust a fenti rekurzív definíciót hűen követve így definiálhatjuk egy képzetbeli programozási nyelven (ahol NIL üres listát, | választást jelöl, Elem pedig bármilyen, már ismert adattípus lehet):

```

TYPE Lista = NIL
  | (feje: Elem, farka: Lista)

```

Az alábbi függvényeljárás egy elemből és egy listából egy elemmel hosszabb listát hoz létre:

```

FUNCTION hozzLétre(e: Elem, l: Lista): Lista;
  RETURN Lista: [e, l];

```

Ennyi bevezető után írd le saját szavaiddal az alábbi függvényeljárások hatását és működését:

A. FUNCTION ismerjFel (I1, I2: Lista): Lista:

```
RETURN IF I1 = NIL THEN I2
      ELSE hozzLétre(I1.feje, ismerjFel(I1.farka, I2))
ENDIF.
```

B. FUNCTION találjKi (e: Elem, I: Lista): Lista:

```
RETURN IF I = NIL THEN I
      ELIF e = I.feje THEN I.farka
      ELSE hozzLétre(I.feje, találjKi(e, I.farka))
ENDIF.
```

Tegyük föl, hogy már léteznek az alábbi (az egyszerűség kedvéért csupa karakterből álló) listák:

I1: (N(E(M(E(S)))))) I2: (T(I(H(A(M(É(R)))))))

Add meg az alábbi műveletek (függvényeljárás-hívások) eredményét:

C. ismerjFel(I1, hozzLétre('+', I2)) = ?

D. találjKi('E', I1) = ?

1992. Második forduló

Első–második osztályosok: Demográfia

A feladat: nyúlpopuláció korcsoportjainak vizsgálata számítógépes szimulációval egy nyúltenyésztő számára.

A szimulációs program először is olvassa be a szükséges paramétereket (12 pont), majd a szimuláció során folyamatosan rajzoljon két grafikont: egyet a nyulak létszámáról az évek függvényében és egy másikat az egyes nyúlkorcsoportok létszámáról a korcsoportok függvényében (25 pont). Például a 9. év elérésekor az alábbi ábrákhoz hasonló ábrákat várunk.

Paraméterként adjuk meg a nyulak számát (max. 1000), maximális életkorukat (10-nél nem nagyobb természetes szám), az egyes nyulak életkorát (természetes szám), a születési, az elsődleges és másodlagos halálozási rátákat, továbbá a járvány valószínűségét (l. alább). A szimulációs program egy-egy lépésében elvégzi mindazt, ami a nyulakkal egy év alatt történhet.

A: A nyulak többsége öregszik és szaporodik, kisebb része a koruktól függő eséllyel (valószínűséggel) meghal. Ezeket az ún. elsődleges halálozási rátákat korcsoportonként 0 és 1 közötti valós számmal adjuk meg, az utolsó szükségképpen 1.

A nyulak (a nemeket nem különböztetjük meg) a koruktól függően szünek utódokat. Ezeket az ún. születési rátákat korcsoportonként egy-egy egész számpárral adjuk meg; a számpár első tagja az átlagos utódszám, második tagja az átlagtól számított maximális eltérés. Az utódszámot egy-egy esetben az

[átlagos utódszám - eltérés, átlagos utódszám + eltérés]

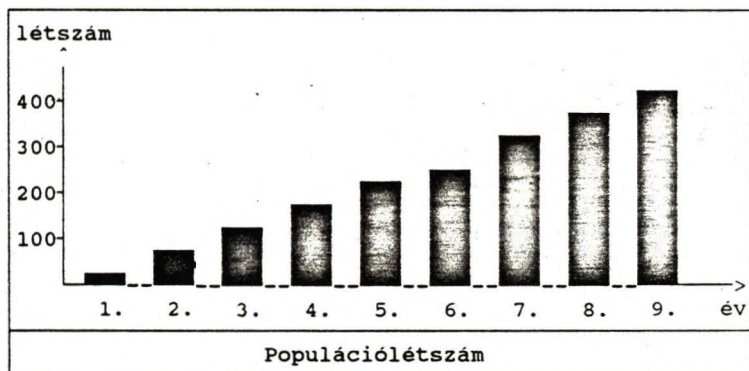
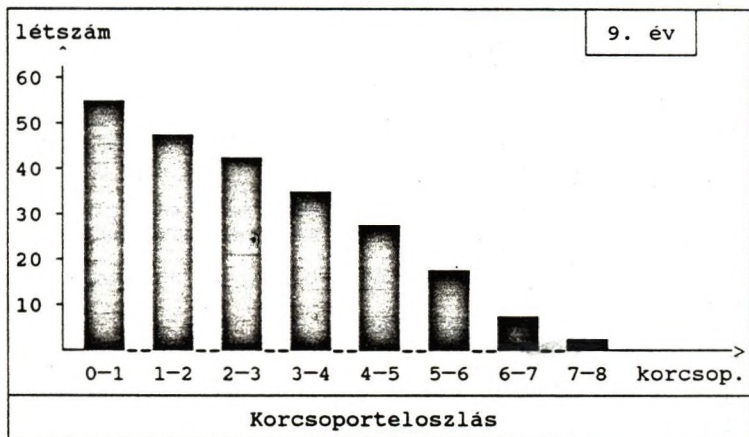
intervallumból vett véletlen egész szám adja meg (21 pont).

B: Egyes években – véletlenszerűen – járvány tör ki a nyulak között; a járvány valószínűsége 0 és 1 közötti valós szám. Ilyenkor a halálozási ráták megnőhetnek. A szimulációs programnak a járvány évében ezeket a másodlagos halálozási rátákat kell alkalmaznia (12 pont).

C: A nyúltenyésztő nem tarthat 1000-nél több nyulat. Ha számuk az 1000-et meghaladja, a korcsoportok létszámát arányosan csökkentve a felesleges nyuszikat megeszzi (10 pont).

A program szerkezetére, olvashatóságára, áttekinthetőségére max. 20 pont kapható.

A két grafikon ilyen lehet (Szlávi P. – Zsákó L.: Szimulációs modellek a populációbiológiában c. könyve alapján):



Harmadik-ötödik osztályosok: Petri-gráf

A feladat: Petri-gráf számítógépes ábrázolása és működésének szimulálása.

A Petri-gráf olyan hálózat, amely csomópontokból, átmenetekből és ezeket összekötő nyilakból áll. A csomópontokban (jelölésük: doboz) lehetnek ún. bigyók (jelölé-

sük: o), tetszőleges számban. A csomópontokból nyilak indulnak ki, amelyek átmenetekhez (jelölésük: dupla vonal) vezetnek. Az átmenetekből újabb nyilak indulnak ki, amelyek csomópontokba vezetnek stb. A hálózat működése a következő:

Egy átmenet aktivizálódásának az a feltétele, hogy az őt közvetlenül megelőző csomópontokban legyen bigyó. Ha az átmenet aktivizálódik, minden befelé vezető nyilon elszív egy-egy bigyót a szomszédos csomópontokból, majd a belőle kivezető nyilakon (a számuk elterhet a befelé mutató nyilak számától) elküld egy-egy bigyót az őt közvetlenül követő csomópontokba. Az átmenetekben a bigyók száma változhat: ha a bejövő nyilak vannak többen, a felesleges bigyók eltűnnek; ha a kifelé vezető nyilak száma a nagyobb, új bigyók keletkeznek. Ha egyidejűleg több átmenet is aktivizálódhat, azaz versenyhelyzet alakul ki, akkor véletlenszerűen egy átmenet aktivizálódik közülük.

Ha a hálózat kezdőállapotát (azaz a bigyók kezdeti eloszlását) ismerjük, meghatározhatjuk azokat az állapotsorozatokat, amelyek a hálózat működését jellemzik. (Több ilyen sorozat is lehet, hiszen a versenyhelyzetben lévő átmenetek véletlenszerűen aktivizálódnak.)

Egy adott állapotból egy másik állapot elérhető, ha van olyan átmenet-aktivizálási sorrend, amelynek eredményeképpen a másik állapot bekövetkezik.

Előfordulhat olyan állapot, amelyben egyetlen átmenet sem aktivizálható. Ekkor a hálózat holtpontra kerül.

A hálózat ciklusba esett, ha

- egy állapot megegyezik valamely korábbi állapottal;
- egy állapot csak abban tér el egy korábbi állapottól, hogy egy vagy több csomópontban több bigyó van, mint korábban volt (ui. ilyenkor végtelen számú bigyó halmozódik fel).

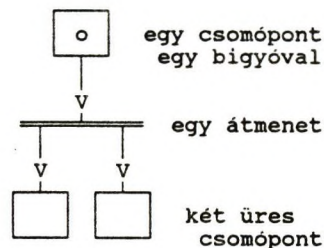
Néhány példa látható a következő oldalon.

Feladatok: (semmi grafika, alfanumerikus be- és kivitel)

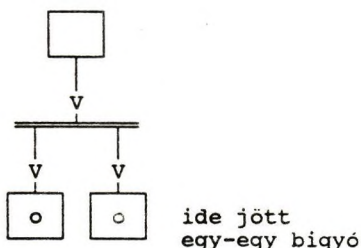
1. Készíts programot Petri-gráf tárolására! A program kezelője adja meg, hogy hány csomópontja és hány átmenete van a gráfnak, és hogy honnan hová vezetnek a nyilak. Csomópontot számjeggyel, átmenetet betűvel jelölünk. A program legyen képes a tárolt gráf szöveges kiírására, de grafikus megjelenítése nem szükséges. Korlátozások: a csomópontok és az átmenetek száma legfeljebb 8-8; nincsenek párhuzamos nyilak (azaz minden átmenet a működése során egy adott csomópontból legfeljebb egy bigyót szív el, ill. egy adott csomópontba legfeljebb egy bigyót küld); a kezdőállapotban egyetlen csomópontban sem lehet egynél több bigyó. (23 pont)
2. A kezdőállapot beolvasása után a program írjon ki egy max. 8 elemű állapotsorozatot (egyet a lehetségesek közül), amelyen a hálózat a működése során áthalad! A program vegye észre, ha a hálózat holtpontra kerül! (27 pont)
3. A kezdőállapot beolvasása és egy másik állapot megadása után a program döntse el, hogy a másik állapot elérhető-e legfeljebb 8 lépésben! Ezt úgy állapítsa meg, hogy az adott kezdőállapotból kiindulva állítsa elő és írja ki az összes elérhető állapotot, és vizsgálja meg, hogy ezek között szerepel-e a keresett állapot. (30 pont)

A program szerkezetére, olvashatóságára, áttekinthetőségére max. 20 pont kapható.

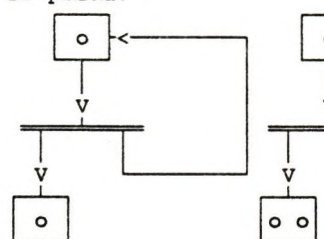
1. példa:



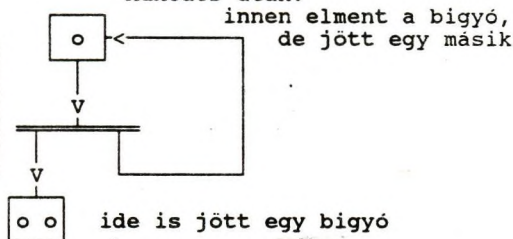
Működés után:



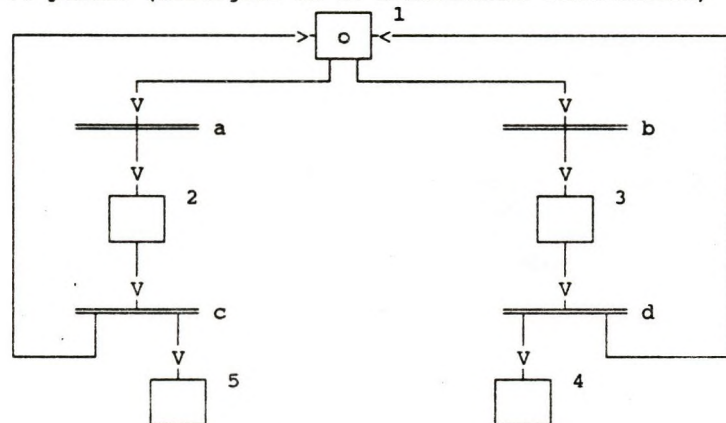
2. példa:



Működés után:



3. példa: (kielégíti az 1. részfeladat feltételeit)



Feltéve, hogy kezdetben az "1" csomópontban van egy bigyó, az "a" vagy a "b" átmenet aktivizálódhat (versenyhelyzet). Ha az egyik aktivizálódik, akkor a másik már nem tud, tehát egy bigyó kerül a "2" vagy a "3" csomópontba. Így vagy a "c", vagy a "d" átmenet válhat aktív a következő lépésben. A "c" vagy a "d" aktivizálása visszaviszi a bigyót az "1" csomópontba, és minden kezdődhet előlről. A "4" és az "5" csomópontokban a bigyók csak gyűlnek-gyűlnek, és azt számlálják, hogy hányszor volt aktív az "a", ill. a "b" átmenet.

Általános Iskolások számítástechnikai versenye, 1991-1992

1991. Első forduló

Összesen: 55 pont

1. feladat (11 pont)

Egy trafikban kétféle tollat lehet vásárolni, az egyik A, a másik pedig B forintba kerül. Készíts programot, amely megmondja, hogy ha C forintért vásároltunk tollat, akkor mennyit vehettünk az első és mennyit a második fajtából! Mondd meg azt is, hogy egyértelműen meghatározható-e az eredmény, és ha nem, akkor az összes megoldást adja meg a programod!

2. feladat (6 pont)

1901. január 1-je hétfői nap volt. Írj programot, amely ennek alapján meghatározza, hogy egy tetszőleges (1901 utáni) év január 1-je milyen napra esik!

3. feladat (12 pont)

Készíts programot, amely beolvas egy egész számot 1 és 99 között, majd kiírja a számot betűkkel (Pl. ha a beolvasott szám 36, akkor azt írja ki, hogy "harminchat").

4. feladat (8 pont)

Feladatunk az első 20 négyzetszám kiírása, amelyre háromféle megoldást készítettünk. Melyik megoldás(ok) helyes(ek), miért?

A) 10 FOR I=1 TO 20
20 PRINT I²
30 NEXT I

B) 10 FOR K=1 TO 20
20 K=K*K: PRINT K
30 NEXT K

C) 10 K=0
20 FOR J=0 TO 19
30 K=K+2*J+1: PRINT K
40 NEXT K

5. feladat (18 pont)

Ismerjük egy labdarúgó-bajnokság eredményeit mérkőzésenként (pl. 1. csapat - 3. csapat: 3:0). Készíts programot, amely táblázatot készít a bajnokság eredményéről a következő formában:

győzelme	döntetlenek	vereségek	lőtt gólok	kapott gólok	pontszám
száma	száma	száma	száma	száma	száma

Győzelemért 2, döntetlenért 1, vereségért pedig 0 pont jár. A táblázatot nem kell rendezni!

1991. Második forduló

Összesen: 70 pont

1. feladat (7 pont)

Készíts programot, amely kiírja a háromjegyű Armstrong-számokat!

Az Armstrong-számokban a számjegyek köbének összege megegyezik magával a számmal (az A köbe = A^3).

2. feladat (11 pont)

A LOGO programnyelv három utasításának jelentése a következő:

REPEAT darabszám [utasítások]

a zárójelbe tett 'utasítások'-at 'darabszám'szor megismétli;

FORWARD lépésszám

'lépésszám' hosszan vonalat húz az aktuális helyzetből kiindulva az aktuális irányban;

LEFT szög

az aktuális irányt balra, a fokokban megadott 'szög'-gel megváltoztatja.

Itt van egy program ezen a nyelven:

```
REPEAT A [FORWARD B LEFT C]
```

Milyen síkidomot rajzol, ha az A , B , C változóknak rendre a következő értékeket adjuk?

1. $A=3$, $B=100$, $C=120$
2. $A=4$, $B=100$, $C=90$
3. $A=360$, $B=1$, $C=1$

3. feladat (13 pont)

Egy lánctalpas jármű lánctalpait be-, ill. kikapcsolhatjuk. A LÁNC(1,1) utasítás az I. lánctalpat bekapcsolja, a LÁNC(1,0) pedig kikapcsolja. A bal oldali lánctalp sorszáma 1, a jobb oldali pedig 2. A FÉNY változó értéke akkor -1, ha a jármű balról érzékel több fényt, akkor 1, ha jobbról, ill. 0, ha balról és jobbról azonos erősségű fényt érzékel. Írj olyan programot, amely a járművet a fényforrás felé vezeti!

4. feladat (12 pont)

Egy H hosszúságú rudat egyenletesen bevonunk N réteggel (a végein is ugyanolyan vastagon, mint az oldalán). $D(I)$ jelenti az I. bevonás utáni átmérőt, $D(0)$ a kezdeti átmérőt. Keres hibákat a következő programrészletben, amely az egyes lépésekben felhasznál anyagok térfogatát határozza meg!

```

100 V=D(0)*D(0)/4*PI*H
110 FOR I=1 TO N
120 U=D(I)*D(I)/4*PI*H
130 PRINT V-U
140 V=V-U
150 H=H+D(I)
160 NEXT I

```

5. feladat (12 pont)

Írd át gyorsabbra (kevesebb, ill. egyszerűbb műveletet végzőre) a következő programrészletet!

```

100 S=0
110 FOR I=1 TO N
120 S=S+A(I)/N
130 NEXT I
140 P=0
150 FOR I=1 TO N
160 P=P+(A(I)-S)^2/N
170 NEXT I

```

6. feladat (15 pont)

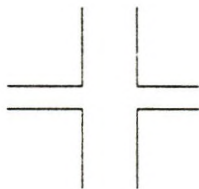
Készíts programot, amely elvégzi a Bongó-sorsolást, majd egy játékos számai alapján megmondja, hogy mennyit nyert az illető!

A Bongón 7 számjegyet sorsolnak ki véletlenszerűen, és a nyereség attól függ, hogy a tippelt számjegyek közül hátról visszafelé olvasva folyamatosan hány egyezik meg a kisorsolt szám számjegyeivel. Ha az utolsó kettő megegyezik, akkor a nyeresmény 70 forint, ha az utolsó három, akkor 700, ha az utolsó négy, akkor 7000 stb.

1991. Harmadik forduló

Összesen: 50 pont

Készíts BASIC programot, amely egy útkeresztvezető forgalmát szimulálja! Az autók a két úton véletlenszerűen érkeznek, az útszakasz végéig haladjanak (ott eltűnnek), és a keresztvezető forgalmát jelzőlámpák irányítsák! Mind a két út egyirányú, irányt változtatni nem lehet.



1991. Országos döntő

1991. április 19.

Összesen: 140 pont

1. feladat (45 pont)

Írj számrendszerek közötti átalakítást végző programot (pozitív egész számok átalakítására), amelyben a felhasználó a következők közül választhat:

- 10-es számrendszerbeli szám kiírása 2-es számrendszerben,
- 2-es számrendszerbeli szám kiírása 10-es számrendszerben,
- 16-os számrendszerbeli szám kiírása 2-es számrendszerben,
- 2-es számrendszerbeli szám kiírása 16-os számrendszerben.

2. feladat (25 pont)

Készíts programot, amely beolvassa egy ember nevét, majd kiírja a monogramját! (Vigyázz a különleges esetekre, pl. Kiss Szabó Zsolt monogramja: K. Sz. Zs.) A név részei, ill. a monogram betűi között egy-egy szóköz van.

3. feladat (25 pont)

Készíts rajzoló programot! A programot a billentyűzetről irányítjuk. Induláskor a kurzor álljon a képernyő közepén, a J(obbra), B(alra), L(e), F(el) billentyűk egyszeri lenyomása jelentsen egységnyi elmozdulást a megfelelő irányban! Így e billentyűk többszöri lenyomásával tetszőleges alakzatok rajzolhatók! Ha a rajzolás során ki-lépünk a képernyőről, akkor lépünk be az ellenkező oldalon!

4. feladat (45 pont)

Készíts programot, amely megadja, hogy egy adott évben mikor van húsvét!

A húsvét a tavaszi napéjegyenlőség (március 21-e) utáni első holdtölte utáni első vasárnap, ill. hétfő. A holdtöltek egymástól 29 és fél napra vannak.

A program olvassa be, hogy

- melyik a keresett év,
 - az adott év első napja a hét melyik napjára esik,
 - hányadikán van az első holdtölte, s aznap délelőtt vagy délután van-e,
- és ezek alapján írja ki húsvétvasárnap, ill. húsvéthétfő dátumát!

Például 1991. január 1-je keddre, az első holdtölte január 30-án délelőttre, a következők pedig február 28-án délutánra és március 30-án délelőttre estek, tehát húsvét az ezt követő első vasárnap, ill. hétfő volt: március 31. és április 1.

1992. Első forduló

Összesen: 100 pont

1. feladat (15 pont)

Készíts programot, amely az $A(N)$ és $B(N)$ vektorokban számjegyenként, a helyértékük növekvő sorrendjében tárolt $(N+1)$ -jegyű egész számokat összeadja, és az eredményt a $C(N+1)$ vektorba teszi!

Például $N=1$, $A(0)=5$, $A(1)=6$, $B(0)=3$, $B(1)=7$ esetén az eredmény $C(0)=8$, $C(1)=3$, $C(2)=1$, mert $65+73=138$.

2. feladat (28 pont)

Készíts programot, amely a grafikus képen egy kurzort mozgat az I, J, L, < kurzormozgató billentyűk hatására! Az aktuális pozícióban megjelenő kurzorképet az 1, 2, 3, 4 billentyűk lenyomásával lehessen bármikor kiválasztani!

Az egyes billentyűkhöz tartozó kurzorképek:

1: U 2: p 3: x 4: +

3. feladat (22 pont)

Egy sakktablára – 8 sora és 8 oszlopa van – elhelyezünk egy bástyát, egy futót és egy huszárt. Készíts programot, amely megadja, hogy üti-e valamelyik bábú a másikat!

A bástya a vele azonos sorban vagy oszlopban levőket üti, a futó a vele azonos átlóban levőket, a huszár pedig a tőle bármely irányban L-alakban levőket, pl.

H
 . . *

4. feladat (21 pont)

Mit rajzolnak a következő, LOGO nyelvű programok? (Induláskor a kurzor a képernyő közepén áll és felfelé néz. Az egyes utasítások jelentése:

LEFT f, RIGHT f	fordulás balra, ill. jobbra helyben, 'f' fokkal;
FORWARD h	előrelépés 'h' egységgel, közben rajzol;
BACK h	hátralépés 'h' egységgel, közben rajzol;
REPEAT db [utasítások]	'utasítások' 'db'-szeri megismétlése.)

- A. RIGHT 150
REPEAT 3 [FORWARD 10 RIGHT 120]
FORWARD 20 BACK 20 RIGHT 60 FORWARD 20
- B. REPEAT 2 [RIGHT 90 FORWARD 10 BACK 10 LEFT 90 FORWARD 10]
RIGHT 90 FORWARD 10
- C. REPEAT 2
[LEFT 90 FORWARD 5 LEFT 90 FORWARD 10 BACK 20
FORWARD 10 RIGHT 90 BACK 5 LEFT 90]

5. feladat (14 pont)

Az $A(6)$ vektorban a pontos idő van (pl. $A(1)=6$, $A(2)=5$, $A(3)=4$, $A(4)=3$, $A(5)=2$, $A(6)=1$ azt jelenti, hogy most 12 óra, 34 perc, 56 másodperc van). Írtunk egy programot, amely, ha meghívjuk, az időt egy másodperccel megnöveli, de nem mindig működik helyesen. Írd le, milyen hibajelenségeket tapasztalnál a használatakor! (Csak tartalmi hibákat keress!)

```
1000 I=1
1010 A(I)=A(I)+1
1020 IF I=1 OR I=3 OR I=5 THEN 1050
1030 IF A(I)<=6 THEN 1100
1040 I=I+1: GOTO 1010
1050 IF A(I)<=9 THEN 1100
1060 A(I)=0: I=I+1: GOTO 1010
1100 ...
```

1992. Második forduló

Összesen: 100 pont

1. feladat (24 pont)

Készíts programot, amely az $A(N)$ és a $B(N)$ vektorokban számjegyenként, a helyértékük növekvő sorrendjében tárolt $(N+1)$ -jegyű egész számokat összeszorozza, és az eredményt a $C(2*N+1)$ vektorba teszi!

Például $N=1$, $A(0)=5$, $A(1)=6$, $B(0)=3$, $B(1)=7$ esetén az eredmény $C(0)=5$, $C(1)=4$, $C(2)=7$, $C(3)=4$, mert $65*73=4745$.

2. feladat (18 pont)

Mit rajzolnak a következő, LOGO nyelvű programok? (Induláskor a kurzor a képernyő közepén áll és felfelé néz. Az egyes utasítások jelentése:

LEFT f, RIGHT f

fordulás balra, ill. jobbra helyben, 'f' fokkal;

FORWARD h	előrelépés 'h' egységgel, közben rajzol, ha kell;
PENUP, PENDOWN	toll felemelése a papírról, ill. leengedése a papírra;
REPEAT db [utasítások]	'utasítások' 'db'-szeri megismétlése.)
A. LEFT 30 REPEAT 3 [REPEAT 2 [FORWARD x RIGHT 60 FORWARD x RIGHT 120] RIGHT 120]	
B. REPEAT 4 [REPEAT 360 [FORWARD 1 RIGHT 1] PENUP RIGHT 90 FORWARD 10 LEFT 90 PENDOWN]	

3. feladat (16 pont)

Készíts verselemző programot, amely egy, a billentyűzetről beolvasott sor szótagjainak hangrendjét adja meg!

Egy szótag magas hangrendű, ha a magánhangzója magas (e, é, i, í, ö, ő, ü, ű), ill. mély hangrendű, ha a magánhangzója mély (a, á, o, ó, u, ú).

4. feladat (18 pont)

A következő programrészlet egy fizikai feladat megoldására szolgál, de nagyon lassan fut. A feladat ismerete nélkül alakítsd át úgy, hogy gyorsabban fusson!

Az átalakítás azon alapuljon, hogy egy ciklus belsejében jobb minél kevesebb műveletet végezni, és hogy a szorzásnál gyorsabb az összeadás, a hatványozásnál pedig a szorzás.

```

100 G=9.81
110 FOR T=0 TO N
120 V(T)=G*T
130 S(T)=1/2*G*T*2
140 NEXT T

```

5. feladat (10 pont)

Egy számítógéppel irányított kisautót helyeztünk el egy labirintusban, a labirintusból való kijutáshoz pedig programot írtunk. A megoldás egy nagyon régi módszeren alapul: haladjunk úgy, hogy a bal kezünk végig a falat érje, így biztosan egyetlen utat sem próbálunk meg kétszer.

Ciklus amíg nem ért ki
 Fordulj balra
 Ciklus amíg fal van előtted
 Fordulj jobbra
 Ciklus vége
 Menj előre egy egységnyit
 Ciklus vége

Ezzel a programmal indítjuk el az autót. Egyes labirintusoknál azt tapasztaljuk, hogy mégsem jut ki belőlük. Milyen felépítésűek ezek a labirintusok?

6. feladat (14 pont)

Az alábbi BASIC programrészlet az A\$ változó alapján állít elő egy szöveget a B\$-ban. Az egyes utasítások jelentése:

RIGHT\$(T\$, i)	a T\$ szöveg jobb szélső i db karaktere,
LEFT\$(T\$, i)	a T\$ szöveg bal szélső i db karaktere,
MID\$(T\$, k, i)	a T\$ szöveg középső i db karaktere a k-tól kezdve,
LEN(T\$)	a T\$ szöveg karaktereinek száma,
CHR\$(i)	az ASCII kódtábla i. jele.

```

1000 S=1: B$=LEFT$(A$,1): A$=A$+CHR$(1)
1010 IF B$=A$ THEN 1200
1020 FOR I=2 TO LEN(A$)
1030   IF RIGHT$(B$,1)=MID$(A$,I,1) THEN S=S+1: GOTO 1090
1040   IF S>2 THEN B$=B$+CHR$(0)+CHR$(S)
1050   IF S=2 THEN B$=B$+RIGHT$(B$,1)
1060   B$=B$+MID$(A$,I,1): S=1
1090 NEXT
1100 ...

```

- A. Mi lesz B\$-ban az 1100-as sorban, ha A\$ tartalma kezdetben
- "Általános iskola"
 - "Fallal bekerített terület"
 - "13333333-szor ismételd meg."
- B. Fogalmazd meg tömören, mit csinál a fenti program!

1992. Harmadik forduló

Összesen: 80 pont

1. feladat (20 pont)

Készíts programot, amely a képernyőre tetszőlegesen háromszöget rajzol, majd vízszintes vonalakkal besatírozza (a háromszög belsejében a képernyő minden második sorában egy-egy vízszintes vonalat húz)! A háromszög csúcspontjainak koordinátáit a billentyűzetről kell beolvasnia.

2. feladat (40 pont)

Készíts programot, amely beolvas egy többsoros szöveget, továbbá a bal és a jobb margó helyét (oszlopszámát) a képernyőn, majd sorokra tagolva kiírja a szöveget a képernyőre a bal és a jobb margó között úgy, hogy ha egy szó nem férne ki teljesen egy sorban, akkor azt már a következő sorban kezdi el! A programnak lehessen 'bal margóhoz igazíts', 'jobb margóhoz igazíts', 'középre igazíts' és 'mindkét margóhoz igazíts' parancsokat adni!

3. feladat (20 pont)

Készíts naptárprogramot, amely beolvassa, hogy 1901. 01. 01. a hét melyik napjára esett, majd egy 1901 utáni tetszőleges év tetszőleges hónapjára kiírja a naptárat, vagyis azt, hogy a kért hónap összes napjának mi a neve!

Jó, ha tudod, hogy a 4-gyel oszthatók általában szökőévek, de a 100-zal oszthatók közül csak azok, amelyek 400-zal is oszthatók.

1985. Első fordulóElső-ötödik osztályosok1. feladat (4 pont)

A szubrutinnak átadott számokat az M1, M2 és M3 változóknban balról jobbra (nem szigorúan) monoton csökkenő sorrendbe rendezi.

Ha szerepel, hogy nem szigorúan monoton csökkenő sorrendbe rendez: 3 pont

Ha csak annyi szerepel, hogy monoton csökkenő sorrendbe rendez: 2 pont

Ha az is szerepel, hogy balról jobbra rendez: +1 pont

2. feladat (5 pont)

Egy természetes szám (≥ 2) prímosztóit írja ki. 5 pont

Ha a ≥ 2 feltétel elmarad, akkor legfeljebb 4 pont adható.

3. feladat (6 pont)

A B() vektorban egyetlen olyan elem sem fordulhat elő kétszer, amely A()-ban is szerepel.

Ha B() minden eleméről kikötik, hogy csak egyszer fordulhat elő, akkor: 4 pont

ha A()-ban sem engedik meg, hogy egy elem kétszer szerepeljen, akkor: 2 pont

ha M = 1 feltételt írnak, akkor: 1 pont

ha A()-ban nem engedik meg az ismétlődést, de B()-ben igen, akkor: 0 pont

4. feladat (6 pont)

- Mindkét program $2^{**}n-1$ és $-2^{**}n$ értékeit nyomtatja ki (ahol ** a hatványozást jelöli). 1 pont

- Mindkét program túlszordulással (OV error) áll meg, mivel nem gondoskodtak arról, hogy L% ne váljék 32767-nél nagyobbá. 2 pont

- Az 1. program már a K% = 16383 és a -K%-1 = -16384 értékek kiírása után megáll, mivel ezt követően (a következő szám kiírása előtt) bekövetkezik a túlszordulás. A 2. program még ki tudja nyomtatni a K% = 32767 és a J% = -32768 értékeket (ez a legnagyobb, ill. a legkisebb ábrázolható egész szám a HT-1080Z gépen), mert túlszordulás csak e számok kiírása után keletkezik. 3 pont

5. feladat (6 pont)

- A THEN után az I index helyett J kell. 2 pont

- A THEN után hibás a csere. 2 pont

- $I = N$ és $J = I$ mellett az $A(J+1)$ hivatkozás hibás (azaz valamelyik ciklus határai hibásak). 2 pont

6. feladat (8 pont)

$$N = 0$$

$$T(0) = -1$$

2 pont

6 pont

7. feladat (10 pont)

Egyenes szakaszt rajzol (összefüggő vonallal) az $(X1, Y1)$ és az $(X2, Y2)$ pontok közé. Helyesen működik, ha a két pont a képernyőn van (HT-1080Z esetén $0 \leq X_i \leq 127$ és $0 \leq Y_i \leq 47$). A 60-as és 70-es sor a két pont koordinátánkénti távolságának maximumát határozza meg H-ban. Ha a távolság 1, akkor 2 pontot kell rajzolni; ha H, akkor H+1-et.

A működés kitalálása: 4 pont

A helyes működés feltétele: 2 pont

A 60-as és a 70-es sor szerepe: 2 pont

H+1 szerepe: 2 pont

8 feladat (15 pont)

1. Ilyenkor B%-ba 0 kerül, ezért a gép az 1000-es és 1010-es sorokat ismételi-geti. 1 pont
2. (Mind a számjegyek, mind a betűk ASCII-kódja követi azok természetes sor-rendjét, ezért a kód a decimális értékből lineáris eltolással valóban elő-állítható. Az ASCII-kódtáblában azonban a számjegyek után nem a betűk követ-keznek, hanem a kettőspont, a pontosvessző, a kisebb-jel stb. Ezért:)
Az 1040-es sorban a szubrutin a számjegyek kódját jól képezi, az A..F betűk kódja helyett azonban a kettőspont, a pontosvessző, a kisebb-jel stb. kódját állítja elő.
A válasz 1 pontot ér, ha csak azt tartalmazza, hogy az előállított kód körül van a hiba, de nem mondja meg, hogy a számjegyek kódja jó, a betűké a rossz.
3. Az 1040-es sor legyen pl. az alábbi:
`1040 IF A% < 10% THEN A% = A% + ASC("0") ELSE A% = A% - 10% + ASC("A")`

2 pont

2 pont

Nem szép, nem elegáns – de jó – megoldásért 1 pont jár.

4. (Az egyes billentyűk hatására 2 megfelelő hatványa, több billentyű egyide-jű lenyomására pedig a hatványok összege kerül a B% változóba. Az 1030-as sor egész osztással az előforduló legnagyobb hatványkitevő értékét számítja ki. Tehát:)
Ha egyszerre több billentyűt ütünk le, akkor közülük mindig a legnagyobb értékűnek megfelelő számot állítja elő a szubrutin az A% változóban. Az "F" billentyű lenyomásakor B%-ba negatív szám kerül, ezért hatása eltér a többi billentyűétől.

Kicsit pontatlan válaszáért 1 pont adható.

5. Ha az "F" billentyűvel együtt egy vagy több másikat is leütünk, a szub-rutin sohasem a neki megfelelő 15-öt írja A%-ba! Ugyanis az "F" hatására a B%-ba íródó $2^{**}15 = 32768$ -at – ahol ** a hatványozás jele – a BASIC interpreter -32768-nak tekinti (a legnagyobb helyértékű bit az előjelbit), és a többi billentyű lenyomásából adódó értéket ehhez adja hozzá.
Ézért a szubrutin a legkisebb értékű lenyomott billentyűnek megfelelő szá-mot állítja elő A%-ban, ha az e fölött lévő összes billentyű le van nyomva, az ez alatt lévők pedig mind el vannak engedve.
Legyen ez az i-edik billentyű ($0 \leq i \leq 14$). Akkor B%-ban $-2^{**}15 + 2^{**}14 + \dots + 2^{**}i = (-2^{**}(15-i) + 2^{**}(14-i) + \dots + 1) \cdot 2^{**}i = -2^{**}i$ lesz, A%-ba tehát i-t tesz a szubrutin.

Példa: F E D C B A 9 8 7 6 5 4 3 2 1 0
X X X X X X X - - - - - - - -

ahol X lenyomott billentyűt, - elengedett billentyű jelöl. A szubrutin a fenti esetben 9-et tesz A%-ba.

Ugyanakkor a szubrutin a legnagyobb értékű elengedett billentyűnek megfelelő számot adja vissza A%-ban, ha az e fölött lévő összes billentyű le van nyomva, és az ez alatt lévők közül is legalább egy le van nyomva. Legyen ez a j-edik billentyű ($1 \leq j \leq 14$).

Ha a j-edik és alatta mind el lenne engedve, B% = $-2^{**}(j+1)$ lenne. (Ez volt az előző eset; most legalább egy be van nyomva a j alattiak közül.)

Ha a j-edik kivételével mindegyik le van nyomva, B%-ba

$$-2^{**}(j+1) + 2^{**}(j-1) + 2^{**}(j-2) + \dots + 2^{**}0 =$$

$$= -2^{**}j - (2^{**}j - 2^{**}(j-1) - \dots - 2^{**}0) = -2^{**}j - 1$$

kerül. Vagyis $2^{**}(j+1) > \text{B\%} > -2^{**}j + 1$,
tehát a szubrutin valóban a j számot írja A%-ba.

Példa: F E D C B A 9 8 7 6 5 4 3 2 1 0
X X X X X X X - - - - X - - - -

ahol X it is lenyomott billentyűt, - elengedett billentyű jelöl. A szubrutin ebben az esetben 8-at tesz A%-ba.

Hibátlan válaszra:

Értékelhető részmegoldásért arányosan kevesebb pont jár.

6. Az új 1020-as és 1030-as sorok a véges, előjeles 16 bites aritmetika egy tulajdonsága miatt működnek jól: $2^{**}14$ és $2^{**}15-1$ közé eső pozitív számok kétszerese negatív!

Ha az "F" billentyűt is leütjük – akár egyedül, akár többedmagával –, B% negatív lesz, és ezért 1030 THEN ágára nem kerül sor, A%-ban 15 marad. Ha "F"-et nem nyomjuk le, akkor B%-ot addig szorozzuk 2-vel (azaz léptetjük a tartalmát balra) és egyúttal csökkentjük egyesével A%-ot, amíg B% negatívvá nem válik. Hogy ez mindenképpen bekövetkezzék, arról az 1010-es sor gondoskodik: addig nem is kerül 1020-ra a vezérlés, amíg B% valamelyik bite nem bil-len be.

Értékelhető részmegoldásért arányosan kevesebb pont adandó.

5 pont
3 pont

1985. Második forduló

Első-ötödik osztályosok: Darazsak

Nincs megoldási-értékelési javaslat.

1986. Első fordulóElső-ötödik osztályosok1. feladat (11 pont)

1.1. B 1.2. A, B, D 1.3. C 1.4. A 1.5. A, E 1.6. B 1.7. A 1.8. B 1.9. D 1.10. C
 Hibátlan megoldásokra jár csak pont (l. a feladat szövegében)!

2. feladat (6 pont)

X tartalma:

7	6	5	4	3	2	1	0
5.	4.	3.	2.	1	1. vonat		

váltó állása: sebessége:
 0 = egyenes 0...7 cm/s
 1 = kitérő

Y tartalma:

7	6	5	4	3	2	1	0
5.	4.	3.	2.	1	2. vonat		

jelző állása: sebessége:
 0 = szabad 0...7 cm/s
 1 = tilos

A két vonat sebességét jelző bitek megadásáért:

2 pont

A váltók állását jelző bitek és jelentésük megadásáért:

2 pont

A jelzők állását jelző bitek és jelentésük megadásáért:

2 pont

Részmegoldásokért 1-1 pont adható.

3. feladat (5 pont)

- 3.1 B A 3.1 kérdésre tökéletes válaszért 3 pont, a C válaszért 2 pont,
 3.2 D értékelhető magyarázatért 1 pont jár.
 A 3.2 kérdésre tökéletes válaszért 2 pont, értékelhető magyarázatért 1 pont jár.

4. feladat (4 pont)

- 4.1 tojáshab – BA, belekeverjük – A, tojás – BP, víz – KA, grízgaluska – KP, üvegesek nem lesznek – A.

4.2 Forró vízben addig főzzük, amíg a grízszemek üvegesek nem lesznek.

- 4.1 2-2 jó válaszonként
 4.2 Jó válasz esetén
 Más válasz is lehet jó!

1-1 pont.
 1 pont.

5. feladat (9 pont)

5.1 A 'hosszú várakozás' gondoskodik arról, hogy a botkormányt bármelyik irányba megnyomva, majd elengedve, a program csak egyetlen megnyomást érzékeljen.

Ha a botkormányt folyamatosan megnyomva tartjuk, a 'rövid várakozás' teszi lehetővé a gyors ismétlést.

5.2 Az 1010-es sor helyesen: A = A AND 15

5.3 Az 1020-1050-es sorokban a következő utasítások helyesek, tetszőleges sorrendben:

```
IF A = 14 THEN PRINT "É";: RETURN
IF A = 11 THEN PRINT "D";: RETURN
IF A = 13 THEN PRINT "K";: RETURN
IF A = 7 THEN PRINT "N";: RETURN
```

- 5.1 Ha csak egy válasz jó, 2 pont, ha mindkettő jó, 3 pont adható.
 5.2 Jó válaszra 3 pont jár. 1 pont adható, ha rájön arra, hogy ez a sor hibás, de nem adja meg a helyes választ.
 5.3 A három hibás utasítás kijavításáért darabonként 1-1 pont jár.

6. feladat (8 pont)

A: $A * A \leq N < (A+1) * (A+1)$, vagyis A egyenlő lesz N négyzetgyökének egészrészével

B: $B = (A+1) * (A+1)$

C: C az (A+1). páratlan szám, azaz $2 * A + 1$

A megadása
 B megadása
 C megadása

4 pont
 2 pont
 2 pont

Rész megoldásokért arányosan kevesebb pont jár.

7. feladat (11 pont)

7.1 A kép szürke pontjai világosabbak lesznek. A kontrasztok élesebbé válnak, mert a közepesen szürke képpontok jobban, a világosabb és a sötétebb képpontok viszont kevésbé világosodnak ki, a tiszta fehér és a tiszta fekete pedig nem változik meg.

7.2 A képet felére kicsinyíti azáltal, hogy minden pontnégyest az átlagokkal helyettesít (az alábbi példában az azonos betűkkel jelölt pontokból lesz egy pont):

```
X X Y Y Z Z   ———>   X Y Z
X X Y Y Z Z
```

7.3 A kép bal felső sarkát 3-szorosra nagyítja azáltal, hogy az egyes pontokat vízszintes és függőleges irányban is megháromszorozza:

$$\begin{array}{cc} X & Y \\ A & B \end{array} \longrightarrow \begin{array}{ccccccc} X & X & X & Y & Y & Y \\ X & X & X & Y & Y & Y \\ X & X & X & Y & Y & Y \\ A & A & A & B & B & B \\ A & A & A & B & B & B \\ A & A & A & B & B & B \end{array}$$

7.1-ért

7.2-ért

7.3-ért

Rész megoldásokért arányosan kevesebb pont adható.

5 pont

3 pont

3 pont

8. feladat (6 pont)

8.1 Egyenlő oldalú háromszög 100 egység oldalhosszal. 1 pont

8.2 Négyzet 100 egység oldalhosszal. 1 pont

8.3 Szabályos 360-szög 1 egység oldalhosszal, azaz majdnem kör. 1 pont

8.4 Egyenlő oldalú háromszög 10 egység oldalhosszal. 1 pont

8.5 Ótágú csillag. 2 pont

Csak teljes értékű válaszokért jár a fenti pontszám!

9. feladat (6 pont)Leggyorsabb: ha az $a()$ vektor eleve növekvően rendezett.Leglassabb: ha az $A()$ vektor eleve csökkenően rendezett.Ha csak az egyik választ adja meg
ha mindkettőt

4 pont,

6 pont.

1986. Második forduló**Első-ötödik osztályosok: Újságszállítás**

Értékelési szempontok:

A1. Előszöri létrehozás; ismételt létrehozás; menürendszer.

15 pont

A2. Mit tesz, ha egy személyt kétszer akarnak felvenni? Ha sorrendben tárol: elejére, közepére, végére lehet-e felvenni? Be kell-e írni az újság nevét vagy "VN"-re veszi fel? Egyszerre egy vagy több beszúrás lehet-e? Városjel-ellenőrzés van-e? Első ember felvétele.

A3. Mit tesz, ha nem létezőt töröl? Mit tesz, ha mindenkit töröl? Lehetséges-e egy vagy több törlés egyszerre? Mi azonosít egy embert: név vagy név-város? Lehet-e törlésből visszalépni "VN" alapján?

A4. Mit lehet módosítani: várost, újságot (nevet nem szabad!)? Módosításkor megadja-e, hogy mi volt a korábbi érték? Kell-e akkor is írni, ha az adatot nem akarja megváltoztatni? Egyszerre egy vagy több módosítás lehetséges?

A1-A4 részfeladatokra

- A5. Rendezett-e a lista? Lapokra tördelés van-e? Lehet-e a lapokat akármeddig nézni? Az utolsót is? 15 pont
Mi van, ha nincs még adat felvéve? Újságneveket ír vagy kódolt értéket? Ír-e fejléceket laponként?
- A6. Lapozás, fejléc; városon belül a nevek névsorban vannak-e? Városok névsorban vannak-e?
- A5-A6 részfeladatokra
- B1. Meg lehet-e adni a városok számát? Útmegadás milyen? Mi van, ha két város között két út van? Megadja-e az A-X útvonalat? 5 pont
- B2. Újságok számának meghatározása jó-e? Eljut-e minden városba? Minden városnál: hány újság ér oda, mennyit visznek tovább és mennyit raknak le? 15 pont
- B3. Végtelen ciklusba kerül-e hurkok miatt? Különböző városokban lévő nyomdákat hogyan kezel? Van-e utalás a legrövidebb útvonalra? 20 pont

1987. Első fordulóElső-ötödik osztályosok1. feladat (12 pont)

$N \geq 1$ és egész

Az $A(N)$ vektor tartalmazzon legalább egy negatív elemet!

F1 az A vektorbeli negatív elemek közül az utolsó, F2 a nemnegatívok közül az utolsó sorszáma lesz. (Ha az utolsót nem mondja, akkor 1 pont levonás.)

A $B(N)$ vektor az A -beli elemek listába fűzésére szolgál a kívánt sorrendben.

A negatívok és a nemnegatívok sorrendje is megfordul – önmagukon belül – az eredetihez képest.

2 pont

4 pont

2 pont

2 pont

2 pont

2. feladat (5 pont)

1

1,3

4

4,2

4,2,5

4,2,5,4

4,2,20

4,2,20,8

4,2,12

4,2,12,6

4,2,2

4,4

0

Tévedésenként 1 pont levonás jár, 4-nél több hiba esetén 0 pont.

3. feladat (12 pont)

X magánhangzók számát

X minden betűje magánhangzó-e

X kódsorrendben (angol ábécé szerinti sorrendben) legnagyobb eleme

4 pont

4 pont

4 pont

4. feladat (9 pont)

A: N négyzetgyökét kétszer számítja be az osztók számába.

az 50-es sorban a feltétel az első és a második osztó megtalálása közötti összes számra teljesül (csak ha az 1 osztó és a $K=1$, akkor lenne szabad az $L=1$ utasítást végrehajtani).

a 80-as sorban, ha N prímszám volt, akkor nincs legkisebb valódi osztó, azaz a kiírást csak $L > 0$ esetben szabad elvégezni.

2 pont

2 pont

2 pont

B: minden olyan számra helyesen működik a program, amelynek a 2 és a 3 osztója, és a szám nem négyzetszám. 3 pont

5. feladat (12 pont)

A: f helyébe -f, vagy 3 pont
minden LEFT helyébe RIGHT és minden RIGHT helyébe LEFT

B: x helyébe -x és f helyébe -f, vagy 3 pont
minden LEFT helyébe RIGHT, minden RIGHT helyébe LEFT, minden FORWARD helyébe BACK és minden BACK helyébe FORWARD

C: x helyébe -x, vagy 3 pont
minden FORWARD helyébe BACK és minden BACK helyébe FORWARD

Ha az A, B, C részfeladatokra mindkét változatot megadja, további 1-1 pont adható.

6. feladat (5 pont)

A: B 2 hatványa és osztója A-nak 2 pont

B: A és B előjele megegyezik 3 pont

7. feladat (10 pont)

Az IGEN címkén fejezi be a végrehajtást, ha B osztója A-nak, egyébként pedig a NEM címkén.

Rész megoldás esetén az IGEN címkén fejezi be, a következő válaszokért:

- A=0 esetén az IGEN címkén fejezi be,
- B=1 esetén az IGEN címkén fejezi be,
- A<B esetén a NEM címkén fejezi be.

8. feladat (8 pont)

A: $F(A) \leq 0$ és $F(B) \geq 0$ 3 pont

B: $F(A) * F(B) \leq 0$ 4 pont

$F(A) = F(B) = 0$ egyszerre nem lehet 1 pont

9. feladat (7 pont)

Néhány lehetséges megoldás:

```

100 INPUT M,H0
110 DIM X(H0),Y(H0)
120 G=9.81
121 MG=M*G
122 E0=MG*H0
130 FOR H=H0 TO 0 STEP -1
140 X(H)=MG*H
150 Y(H)=E0-X(H)
180 NEXT H

```

```

100 INPUT M,H0
110 DIM X(H0),Y(H0)
120 G=9.81
121 MG=M*G
122 EH=MG*H0
123 EM=0
130 FOR H=H0 TO 0 STEP -1
140   X(H)=EH
150   Y(H)=EM
160   EH=EH-MG
170   EM=EM+MG
180 NEXT H

```

Pontszám = 6 – a ciklus belsejében levő szorzások száma, ill. +1 pont, ha a ciklusmagban az értékadások, összeadások, kivonások, tömbindexelések együttes száma ≤ 8 .

10. feladat (15 pont)

A: A beolvasott számok bináris alakban

240 = 1111 00 00		^ balra	
242 = 1111 00 10			Az utolsó két bit változik,
243 = 1111 00 11			a többi változatlan.
241 = 1111 00 01	V jobbra		
243 = 1111 00 11		^ hátra	
251 = 1111 10 11			A következő két bit változik,
255 = 1111 11 11			a többi változatlan.
247 = 1111 01 11	V előre		

A beolvasott byte felső 4 bitje állandóan 1.

Az alsó két bit az X irányú, a következő két bit az Y irányú elmozdulástól függő kódszót tartalmaz.

Az elmozdulás irányát a kódok sorrendje adja meg.

- B: (X,Y)= az egér pozíciójának koordinátái. 1 pont
 (X1,Y1)= az egér pillanatnyi állapotának 2-2 bites kódja. 1 pont
 (X0,Y0)= az egér előző állapotának 2-2 bites kódja. 1 pont
 (X9,Y9)= az előző és a pillanatnyi állapotból számított decimális érték. 1 pont
- C: A program követi az egér mozgását és pontsorozatként felrajzolja a bejárt utat a képernyőre. 1 pont
 A képernyőn a (10,10) pontból indul és nem ellenőrzi, hogy a kirajzolandó pont a képernyőre esik-e. 1 pont
- D: feltétel1: $X_9=2 \text{ OR } X_9=23 \text{ OR } X_9=31 \text{ OR } X_9=10$ (jobbra) 3 pont
 feltétel2: $X_9=20 \text{ OR } X_9=32 \text{ OR } X_9=13 \text{ OR } X_9=1$ (balra)
 feltétel3: $X_9=2 \text{ OR } X_9=23 \text{ OR } X_9=31 \text{ OR } X_9=10$ (előre)
 feltétel4: $X_9=20 \text{ OR } X_9=32 \text{ OR } X_9=13 \text{ OR } X_9=1$ (hátra)

Ha egy feltételt hibátlanul ad meg:

Ha az ellenkező feltételt is hibátlanul adja meg:

A további két feltétel megadásáért feltételenként:

+2 pont

+1 pont

1987. Második forduló

Első-ötödik osztályosok: Üzenetközvetítő

Értékelési szempontok:

- | | | |
|---------------------------------------|---|--------|
| 1. Vezérlő rész: | | |
| 1.1. | Bejelentkezés | 3 pont |
| | - nemlétező név beengedése: | |
| | -1 pont | |
| | - nemlétező név osztály beengedése: | |
| | -1 pont | |
| | - nem ismer minden személyt: | -1 |
| | pont | |
| 1.2. | Menü vagy kérdések sorozata | 5 pont |
| | - minden funkció használható-e: | 3 pont |
| | - ugyanaz-e Kiss Attilának, ha nem: | 2 pont |
| | - beenged-e illegális funkcióba: | -1 |
| | pont | |
| | - lehet-e bejelentkezés nélkül bármit tenni: | -1 |
| | pont | |
| 1.3. | Dátumbeírás | 2 pont |
| | - ha beírása előtt lehet valamit csinálni: | -1 |
| | pont | |
| 2. Összes levél kiírása | | |
| 2.1. | Lapokra tagolás levelenként | 2 pont |
| | - rosszul működik, ha nincs levél: | -1 |
| | pont | |
| | - utolsó levél nem olvasható: | -1 |
| | pont | |
| | 2.2. Kijelzendő: címzett, küldő, tartalom, törlési fajta. | 2 pont |
| 3. A dolgozónak szóló levelek kiírása | | |
| 3.1. | Levelenkénti lapokra tagolás | 2 pont |
| 3.2. | Megkapja a személyesen neki szólókat | 2 pont |
| 3.3. | Megkapja az osztályának szólókat | 2 pont |
| 3.4. | Megkapja a vele azonos beosztásúaknak szólókat | 2 pont |
| 3.5. | Megkapja a mindenkinek szólókat | 2 pont |
| 3.6. | Megtudja, hogy kitől kapta a levelet | 2 pont |
| | - ha másnak szóló levelet is kap: | -4 |
| | pont | |
| 4. Levél küldése | | |
| 4.1. | Csak olyannak küldhet, akinek szabad | 4 pont |
| 4.2. | Ellenőriz törlési kapcsolatokat (körlevelet ne törölhes-
sen a címzett és ne töröljön egy olvasás után; más kap-
csolatot nem kell tiltani, de nem baj, ha esetleg tiltják) | 2 pont |
| 4.3. | Minden fajta levél létezik | 5 pont |
| 4.4. | Minden törlési mód létezik | 4 pont |
| 4.5. | Egymás után több levelet is küldhet | 1 pont |
| 4.6. | Ki lehet lépni küldés nélkül | 1 pont |
| 5. Levél törlése | | |
| 5.1. | Egy olvasás után törlődik-e | 2 pont |
| 5.2. | Adott dátumkor törlődik-e | 3 pont |

5.3.	A küldő visszavonhatja		3 pont
	- ha más is visszavonhatja:	-1	
	pont		
	- ha a sikertelen törlésről nincs visszajelzés:	-1	
	pont		
5.4.	A kapó törölheti		2 pont
	- ha a sikertelen törlésről nincs visszajelzés:	-1	
	pont		
5.5.	Kiss Attila törölhet-e bármit		2 pont
	- ha a sikertelen törlésről nincs visszajelzés:	-1	
	pont		
5.6.	A törlő megtudhatja-e, hogy milyen levelet vonhat vissza		2 pont
5.7.	Ki lehet-e lépni törlés nélkül		1 pont
6.	Adatszerkezet		
6.1.	Levél: CÍMZETT (kódolva), KÜLDŐ (kódolva), TARTALOM, TÖRLÉSI MÓD, DÁTUM, LEVELFAJTA (ha a címezettből nem derül ki)		6 pont
6.2.	Dolgozó: NÉV, KÜLDÉSI JOG, BEOSZTÁS		3 pont
6.3.	Osztály: NÉV, KÓD		1 pont
6.4.	Levelek tárolása (lista, mutató, gyors művelet)		2 pont
7.	Programszerkezet		
7.1.	Eljárásokra tagolás, modularitás		3 pont
7.2.	Közös funkciók kiemelése egy eljárásba		2 pont
7.3.	Megjegyzések, a programszöveg tagolása		2 pont
7.4.	Egyszerűség, világosság		2 pont
7.5.	Értelmes változónevek		1 pont
8.	Felhasználói tulajdonságok		
8.1.	Kérdések szövegezése		1 pont
8.2.	Hibajelzések szövege, láthatósága		1 pont
8.3.	Csak olyan funkciót lát el a programból, amit használhat is		2 pont
8.4.	Törlést csak törölhetőre kérdez		1 pont
8.5.	Felesleges dolgot (pl. osztályt) nem kérdez		1 pont
8.6.	A képernyőkezelés esztétikus		1 pont
8.7.	Egyszerűen lehet választani a lehetőségek közül (menü)		3 pont
	- ha teljes szavakat kell beírni:	-1	
	pont		
	- következetes beolvasás hiányzik:	-1	
	- nem szólít fel billentyű lenyomására:	-1	
	pont		

1988. Első fordulóElső-ötödik osztályosok

Összpontszám: 75

Beküldési
határ: 30 pont

Postázási határidő: 1988. február 2.

1. feladat (6 pont)

- A $b^{2-4} \cdot a^c$ diszkriminánsban i növekedésével b^{2-4} értéke sokkal gyorsabban nő $4 \cdot a^c$ értékénél, és a számábrázolás pontatlansága miatt a b^{2-4} mellett a $4 \cdot a^c$ előbb-utóbb elhanyagolható értékűvé válik.
Ekkor $x_2 = -b + \sqrt{b^2} = 0.0$ lesz az eredmény a várt 1.0 helyett! 1 pont.
- A 24 bites mantisszával max. $2^{23} = 8\,388\,608$ -at, azaz 10^6 nagyságrendű számokat lehet pontosan ábrázolni. 1 pont.
- A kivonáshoz a kitevőknek egyenlőknek kell lenniük, ezért $i > 6$ esetén már fellép az említett jelenség. 1 pont.
- Az adott számábrázolás mellett max. $2^{(2^7)} = 2^{128} \approx 10^{38}$ nagyságrendű számok ábrázolhatók. 1 pont.
- Ezért túlszordulás $b > 10^{19}$, vagyis $i > 19$ esetén lép fel. 1 pont.

2. feladat (15 pont)

- 2.1.[1] meghatározza azt a hossz-t, amely mellett az $[1, \text{hossz} - 10]$ intervallumba eső számok összege még kisebb, az $[1, \text{hossz}]$ intervallumba eső számok összege pedig már nagyobb m -nél. 1 pont
- [1] eredményéből kiindulva [2] azt a hossz-t határozza meg, amely mellett az $[1, \text{hossz}]$ intervallumba eső számok összege még éppen kisebb, az $[1, \text{hossz} + 1]$ intervallumba eső számok összege pedig már éppen nagyobb m -nél. 1 pont
- [2] eredményéből kiindulva [3] azt vizsgálja, hogy [az első hossz szám összege, m] intervallum felbontható-e hossz számú, egyenlő hosszúságú részintervallumra. Ha igen, megvan a megoldás, hiszen csak e részintervallumok hosszával kell megnövelni az $[1, \text{hossz}]$ intervallum minden elemét. Egyébként hossz-t 1-gyel csökkentjük, és folytatjuk. 2 pont
- 2.2.4095 esetén egyszer sem, ugyanis 4095 éppen az első 90 természetes szám összege, és ezért [3] ismétlési feltétele már kezdetben hamis. 1 pont
- 4096 esetén 89-szer, ugyanis 4096 2 hatványa, amelynél a keresett intervallum hossza 1 (l. a 2.4. választ). Mivel a [2] ismétlés végén hossz=90, a hossz=1 érték megtalálásáig éppen 89-szer kell megismételni a törzset. 1 pont
- 2.3. Itt három ismétlés van, amelyekben – elvileg – elakadhatna a program. De [1] biztosan befejeződik, mert hossz-at 10-esével növelve az első hossz szám összege előbb-utóbb biztosan nagyobb lesz a véges m -nél (vagy túlszordulás lép fel). 1 pont
- De [2] is befejeződik, mert hossz-at 1-esével csökkentve legfeljebb 9 lépésben m -nél megint kisebb lesz az első hossz szám összege. 1 pont

- Végül [3] is biztosan véget ér, hiszen a legrosszabb esetben is hossz=1-gyel minden szám maradék nélkül osztható. 1 pont
- 2.4. Ha m felbontható az $[i, j]$ intervallumba eső számok összegeként, akkor $m = \text{hossz} \cdot (i+j)/2 = \text{hossz} \cdot (2^i + \text{hossz} - 1)/2$, hiszen $j = i + \text{hossz} - 1$. A vizsgálandó esetben $m = 2^{**}k$ alakú (ahol $**$ a hatványozás jele), vagyis tetszőleges k -ra teljesülnie kell a következő egyenlőségnek: $2^{**}(k+1) = \text{hossz} \cdot (2^i + \text{hossz} - 1)$. De ha hossz páros, akkor $2^i + \text{hossz} - 1$ páratlan, és fordítva, tehát mindkét tényező egyszerre csak hossz=1 mellett lehet 2 harványa. 3 pont
- 2.5. A program gyorsabbá tehető, ha a szorzások és az osztások helyett összeadást és kivonást végzünk. Erre van lehetőség, ugyanis az első hossz szám összegé-nél az első hossz+1 szám összege éppen (hossz+1)-gyel nagyobb. Ha bevezetünk egy változót az összeg tárolására, akkor ennek értékét az [1], [2] és [3] ismétlésekben minden lépésben meghatározott értékkel kell növelni, ill. csökkenteni. 2 pont.
- Ha az adott programozási nyelven ábrázolható legnagyobb egész számot ismerjük, akkor tudunk felső korlátot adni a hossz-ra, és így az [1] és [2] ismétlés helyett a gyorsabb bináris (logaritmikus) keresést használhatjuk. 1 pont.

3. feladat (8 pont)

- 3.1. A fekete szemek száma 1-esével nő, ha két fehéret húzunk, egyébként, 1-esével csökken. 1 pont
A fehér szemek száma 2-esével csökken, ha két fehéret húzunk, egyébként nem változik. 1 pont
- 3.2. A fehér szemek számának párossága (ez ún. invariáns tulajdonság)! Ha kezdetben páratlan volt a fehér szemek száma, akkor végig páratlan is marad, ha páros volt, akkor páros is marad. 3 pont
- 3.3. Minden lépésben 1-gyel csökken a kávészemek száma a dobozban. 1 pont
A fehérek kezdeti számától függ, hogy milyen színű lesz az utolsó: ha páratlan, akkor fehér; ha páros, akkor fekete. 2 pont

4. feladat (9 pont)

- 4.1. A példákban előlől hátrafelé kiolvashatók a következő kódok: 2 pont
0=0000, 1=1000, 2=0100, 3=1100, 5=1010, 6=1101, 7=1111, 9=1001.
Visszafele olvasva néhány kód más: 1=0001, 2=0010, 3=0011, 5=0101, 6=1011. 1 pont
Hiányzik a 4-es és a 8-as számjegy, amelyekhez a még hiányzó 0110, 0111 és 1110 kódszavak rendelhetők. 1 pont
A hozzárendelés a 4.2 választól függ. 1 pont
- 4.2. A számjegyek kódjából ki lehet deríteni az olvasási irányt, és így helyre lehet állítani az eredeti sorrendet. 2 pont
Ehhez az szükséges, hogy a négy közül legalább egy számjegynek ne legyen szimmetrikus a kódja: ezért írjuk elő, hogy a legnagyobb helyértékű jegy csak 1, 2, 3, 4, 5 és 6 lehet. 2 pont
Ebből következik, hogy a 4 számjegynek két kódja van: 0111 és 1110, a 8-nak pedig egy: 0110. 2 pont

5. feladat (6 pont)

- A beolvasott decimális számot normalizálja. 1 pont
 A számot b db számjeggyel írja ki. 1 pont
 A többi számjegyet kerekíti. 1 pont
 Ha a \$ 0-val kezdődik, akkor hibásan normalizál. 1 pont
 Ha az utolsónak kiírandó számjegy 9-es lenne, és ezt a kerekítés miatt meg kell növelni, akkor nem számjegyet ír ki, hanem a 9-es után következő kódú karaktert. 2 pont

6. feladat (7 pont)

- 6.1. A és B másképpen működik, ha a számsorozat 1-esekkel kezdődik vagy végződik. 1 pont
 Ekkor csak B veszi figyelembe az első sorozat hosszát. 2 pont
 6.2. Mindkettő hibás, ha a számsorozat 1-esekkel végződik. 1 pont
 Ekkor B kiszámolja, de nem írja ki a megoldást. 2 pont
 Ekkor A-ban a belső ciklus feltételében indexhatártúllépés lesz. 1 pont

7. feladat (7 pont)

- A 90-es sorbeli RAJZOLÁS a koordináták által kijelölt négyszöget rajzolja a képernyőre. 2 pont
 A 110-es sorbeli RAJZOLÁS hasonló, de ALFA szöggel elforgatott négyszöget rajzolja a képernyőre. 4 pont
 Az utóbbi a (0,0) pont mint középpont körül van elforgatva. 1 pont

8. feladat (9 pont)

- A program befejeződik, ha olyan pontot talál, amely minden szomszédjánál magasabb, de mégsem a legmagasabb. 2 pont
 Ha fennsíkra ér, végtelen ciklusba kerül. (A fennsík egynél több azonos magasságú pontból áll, amelyeket alacsonyabb pontok vesznek körül.) 3 pont
 Síkon haladva mindig "délkeleti" irányba törekszik. 1 pont
 Akkor is végtelen ciklusba kerül, ha egy síkon haladva alacsonyabb pontokba ütközik, vagy eléri a térkép szélét. 2 pont
 1 pont

9. feladat (8 pont)

- 9.1. Törtvonalban vagy egyenes vonalban halad a fényforrás felé. 2 pont
 De ha a fényforrás az autó mögött van, távolodik tőle. 1 pont
 9.2. Az e a fényerősség-különbségre vonatkozó érzékenységi küszöb. 1 pont
 9.3. Ha e olyan kicsi, hogy az autó a lehető legkisebb elfordulás után is azonnal az ellenkező irányba akar fordulni (azaz nem tud elég kicsit fordulni). 2 pont
 Ha e olyan nagy, az autó állandóan elkerüli a fényforrást. 1 pont
 Ha a fényforrás olyan gyorsan mozog, hogy a játékautó nem tud utánafordulni. 1 pont

1988. Második fordulóElső-ötödik osztályosok: Értelmező program

Értékelési útmutató

Összesen
84 pont

Alapfunkciók:	28 pont
M	- 5
F	- 2
N, R	- 1
T	- 1
H	- 2
I	- 10 (I-ben eljárás hívás: 3, a többiek: 7 pont)
V	- 2 (alapállapotba állítás)
szakasz	- 5 ("nem folytonos" esetben csak 2 pont)
Eljárásfunkció:	19 pont
E	- 7 (időleges eljárásdefiniáció: csak 4 pont)
D	- 2
hívás	- 10 (eljárásban I: 3, a többiek: 7 pont)
Vezérlőrész:	7 pont
beolvasás	- 2 (nem teljes programbeolvasás esetén: 0)
végrehajtás vezérlése	- 5
Szerkezet:	17 pont
adatszerkezet	- 5
strukturáltság	- 5 (eljárások, algoritmusszerk. használata)
tagoltság	- 3 (eljárások, algoritmusszerkezetek formája)
magyarázatok	- 2
fejleszthetőség	- 2
Bővítések:	13 pont
(hibakezelés esetleg	- +8)
utasításnév	- 1
paraméter	- 1
ismétlés	- 1 (zárójelhiba)
eljárásdefiniáció	- 2 (zárójelhiba, már létező újradefiniálása)
fenntartott név	- 1
kimegy a képről	- 1
hibás törlés	- 1 (Dobd el utasításban)
értelmes hibajelzés	- 3
teljes utasításnév	- 2

1989. Első fordulóElső-ötödik osztályosok

Összpontszám: 52 pont

Beküldési
határ: 20 pont

Postázási határidő: 1989. január 31.

1. feladat (3 pont)

Az eredmény 64, mivel a 20-as és az 50-es sori egyszer, a 60-ast kétszer, a 70-est négyszer s.i.t. hajtja végre a program.

Jó eredmény:

1 pont

Magyarázat:

2 pont

2. feladat (8 pont)

1030 DO WHILE J >= 1 AND IS\$ < T\$(J)

1 pont

1040 T\$(J+1) = T\$(J): J = J - 1

1 pont

1060 T\$(J+1) = IS\$

1 pont

1.---> Az (I-1)-edik elemig a T\$() vektor rendezett,
és IS\$ a beszúrandó:

1 pont

2.---> A (J+1)..I intervallumban a T\$() vektor rendezett,

1 pont

IS\$ helye az 1..J indexű elemek valamelyike:

1 pont

3.---> Az 1..I intervallumban rendezve van a T\$() vektor:

1 pont

3. feladat (6 pont)

- az x tizedestörtöt kettedestörtté (azaz kettes számrendszerbeli törtté alakítja) 1 pont
- befejeződik, ha már előállítottunk 30 számjegyet, 1 pont
- vagy ha a végtelen szaka-szos kettedestört első ismétlődő szaka-szának vége van, 2 pont
- vagy ha nulla maradékot kapunk valamelyik lépésben, azaz az utolsó értékes jegynél 1 pont
- $x=0.3$ -ra az eredmény 0.01001 1 pont

4. feladat (6 pont)

- K1\$ kulcs szerinti helyére igyekszik tenni a (K1\$,K2\$) sort a táblázatban 2 pont
- ha az így kiszámított hely szabad, akkor oda teszi 1 pont
- ha nem, akkor megkeresi a mögötte levő legközelebbi üres helyet, és odateszi 2 pont
- ha nincs üres hely, akkor kiírja, hogy "baj!!!" 1 pont

5. feladat (6 pont)

A D() vektor tartalmazza az egyes érmék darabszámát.

PénzviSSzaadás(ÖSSZEG):

Ciklus l = 1-től 5-ig

Ha ÖSSZEG >= FT(l) akkor DB := INT(ÖSSZEG / FT(l))

Ha D(l) < DB akkor DB := D(l)

D(l) := D(l) - DB

Ki: DB; " db. "; FT(l); " Ft-os"

ÖSSZEG := ÖSSZEG - DB * FT(l)

Elágazás vége

Ciklus vége

Ha ÖSSZEG > 0 akkor Ki: "Nem tudok visszaadni"

Eljárás vége.

DB korlátozása

2 pont

D(l) csökkentése

2 pont

Visszaadás lehetetlenségének kezelése

2 pont

6. feladat (7 pont)

- A program egy egyszerű hóembert rajzol: három kört egymás tetején. (Egymás melletti körök – fekvő hóember – csak 2 pontot ér.) 3 pont
- A felső két kör sugara mindig kétharmada az alatta levő kör sugarának. 1 pont
- A legalsó kör sugara 30 egység. 1 pont
- A GVONAL eljárás egy félkörívet közelít szabályos sokszöggel. 1 pont
- Az :Y paraméter a körív sugarát adja meg. 1 pont

7. feladat (8 pont)

- A G1 0-ba ragadásának felismeréséhez A=0, B=1, C=1 szükséges, így a 100-as sor első száma: 6. 1 pont
- Ha a kapu jól működik, akkor Y=0, egyébként Y=1, tehát a második szám: 0. 1 pont

A másik két sor hasonlóan:

2 pont

- 110-es sor: 1,0 vagy 3,0 vagy 5,0 a jó számpár.

2 pont

- 120-as sor: 7,1 a jó számpár.

2 pont

Indoklás:

A tesztelt kapu kimenetének hibája akkor deríthető ki, ha a kapu

1 pont

vezérlése olyan, hogy a kimeneten 1-nek kellene lennie,

és ez az 1-es ki tud jutni az áramkör Y kimenetére.

1 pont

(Például a G1 kimenete akkor lesz 1, ha A=0 és B, C mindegyike 1, mivel ekkor G1 második bemenete 0 lesz. A jel akkor jut ki G3 kimenetére – ami az áramkör kimenete – ha G3 másik bemenete 0, ez itt teljesül. A hiba tehát kideríthető a fenti vezérléssel, mert G1 hibája esetén 0 helyett 1 jelenik meg az áramkör kimenetén. Itt természetesen kihasználtuk azt, hogy az áramkörben egyszerre csak egy hibás kapu lehet.)

8. feladat (8 pont)

Problémánként 1-1 pont (csak algoritmikus problémákra adható), néhányat felsorolunk:

- a ki-, ill. beszállást 10 másodpercen belül be kell fejezni,
- a lift, miközben egy szinten áthalad, egyetlen hívást és egyetlen parancsot képes fogadni (hiába nyomnak meg egyszerre több gombot),
- nem veszi észre, hogy az adott szintre már hívták (ugyanazt többször is megnyomhatják),
- nem veszi észre, hogy a többszörösen hívott szintekre nem kell még egyszer elmennie,
- nem áll meg közbülső szinteken,
- 32-nél több hívás esetén indexhatár-túllépéssel leáll,
- ha az utasok utolsó, liftben maradt csoportja nem száll ki az általuk megjelölt szinten, akkor az ajtó becsukódása után mindaddig a liftben kénytelenek maradni, amíg kívülről valaki nem hívja a liftet,
- beszállás után, ha a liftnak nincs több tárolt úticélja, akkor az utasoknak a következő lifthívást meg kell várniuk, mert a lift nem figyel a PARANCS gombot.

1989. Második fordulóElső-ötödik osztályosok: Digitális áramkör szimulációja

Értékelési szempontok:

Osszpontszám: 100 pont

1. Adatszerkezet:

1.1.	kapufajták táblázata	3 pont
1.2.	kapuk táblázata	2 pont
1.3.	kapubemeneti jelek táblázata	1 pont
1.4.	kapukimeneti jelek táblázata	1 pont
1.5.	csomópontok táblázata	2 pont
1.6.	eseménylista (változások követése)	3 pont

2. Hálózat beolvasása

2.1.	a kapuneveket és csak azokat lehet beolvasni	4 pont
2.2.	minden kaput megfelelő be- és kimenetszámmai lehet beolvasni	4 pont

3. Működés

3.1.	az egyes kapuk helyesen működnek	12 pont
3.2.	jelek terjednek a következő kapu bemenetére	4 pont
3.3.	a kimenetre kijutnak a jelek	2 pont
3.4.	leáll, ha már nincs a kapuk kimenetein változás	6 pont
3.5.	visszacsatolásos hálózatot is tud kezelni	4 pont
3.6.	bemeneti kombinációk beolvasása	2 pont

4. Összes bemenet előállítása

4.1.	jól előállítja	8 pont
4.2.	a képernyőről nem futnak ki a szimuláció eredményei (egyesével állítja elő vagy a képernyő betelésekor lehet lapozni)	2 pont

5. Ellenőrzések

5.1.	egy kapu kimenetét sehova sem kötötték, és az nem is az áramkör kimenete	2 pont
------	--	--------

5.2.	egy kapu bemenetét sehova sem kötötték, és az nem is az áramkör bemenete	2 pont
5.3.	két (vagy több) kapu kimenetét összekötötték	4 pont
5.4.	van olyan kapu a hálózatban, amelyet a bemenetről nem lehet elérni	8 pont
5.5.	az áramkör valamelyik kimenete nem valamelyik kapu kimenete	1 pont
5.6.	az áramkör valamelyik bemenetét nem kötötték egyetlen kapu bemenetére sem	1 pont
5.7.	a korlátnak állított időegységszám letelte után sem alakul ki stabil jel az áramkör kimenetén	2 pont
6.	Programminőség	
6.1.	képernyőtörítés, címkíírás a megfelelő helyeken	2 pont
6.2.	barátságos beolvasás, kiírás	4 pont
6.3.	magyarázatok a programszövegben	2 pont
6.4.	bekezdéses leírás	2 pont
6.5.	függőleges tagolás a programszövegben (pl. egyes részek között üres sor)	2 pont
6.6.	eljárásokra bontás világos szerkezetű	4 pont
6.7.	rövid programsorok	1 pont
6.8.	értelmes változónevek	1 pont
6.9.	menürendszer	2 pont

4. feladat (13 pont)

- A: A (*2*) és a (*4*) változat kever szabályosan. max. 6 pont
(6 pont a hibátlan válaszra. Ebből 2-2 pont levonás minden tévedésre; 0 pont, ha kettőnél többet téved.)
- B: Az (*1*) és a (*3*) változat egyes lapokat megtöbbszörözhet, másokat kihagyhat. 2 pont
Az (*1*) változat nem jegyzi meg, hogy mely lapokat osztotta már ki. 2 pont
A (*3*) ezt ugyan megteszi, de a már egyszer kiosztott lapokból is újra oszt. 2 pont
- C: A (*2*) változat a VÉLETLEN() függvényt a pakli leosztása közben várhatóan egyre gyakrabban hívja meg egy-egy lap kiosztásához, ugyanis egyre valószínűbb, hogy a függvény egy már kiosztott lap számát állítja elő. 2 pont
(Háttérinformáció a javító tanárnak: Az n-edik lap kiosztásához szükséges hívások száma geometriai eloszlást követ, amelynek várható értéke a kívánt esemény valószínűségének reciproka, esetünkben $32/(33-n)$. Ezért a teljes program lefutásához várhatóan $32/32 + 32/31 + 32/30 + \dots + 32/1$ hívás szükséges. Megsejtése plusz pont!)

5. feladat (7 pont)

Nem csökkenő sorrendbe rendezve kiír 1 és 300 közötti egész számokat.

- Rendezve kiír: 3 pont
növekvően: 2 pont
1 és 300 közötti egészeket: 2 pont

6. feladat (8 pont)

- A: SZÓ = HIDEG 3 pont
B: KULCS = K 1 pont
C1: A művelet KÓD(I-1) operandusából és KÓD(I) eredményéből kiszámítható SZÓ(I) operandusa, és így a szó hátralőről visszafelé haladva helyreállítható, 2 pont
- vagy
C2: az itt alkalmazott XOR művelet asszociatív és kommutatív; ha két operandusa azonos, akkor 0 az eredménye; ha egyik operandusa 0, akkor az eredménye azonos a másik operandusával. Ezért $SZÓ(I) = KÓD(I)$ művelet $KÓD(I-1)$, ami az első betű kivételével mindig igaz. Az első betűre pedig $KULCS = KÓD(0) = SZÓ(1)$ művelet $KÓD(1)$ igaz. 4 pont

Harmadik-ötödik osztályosok

Összpontszám: 87 pont

Beküldési
határ: 35 pont

Postázási határidő: 1990. január 31.

1. feladat (15 pont)

A:	3 4 5	5 pont
	2 1 6	
	9 8 7	
B:	6 3 8	5 pont
	2 1 4	
	7 5 9	
C:	9 3 8	5 pont
	2 1 4	
	6 5 7	

2. feladat (6 pont)

Az egységkör területét, vagyis pi-t (csak 2 pont, ha negyedkör területe a válasz).

3. feladat (11 pont)

A: helyes		1 pont
	(1), (5), (9), (6), (12)	2 pont
B: hibás		1 pont
C: helyes		1 pont
	(1), (3), (8), (10), (6), (13)	2 pont
D: helyes		1 pont
	(2), (6), (13), (3), (8), (10)	2 pont
E: hibás		1 pont

4. feladat (10 pont)

HIBA 1 -)-jel a (-párja nélkül	2 pont
HIBA 2 -)-jel le nem zárt [-jellel	2 pont
HIBA 3 -]-jel a [-párja nélkül	2 pont
HIBA 4 -]-jel le nem zárt (-jellel	2 pont
HIBA 5 -	nincs minden (-jelnek)-párja	1 pont
HIBA 6 -	nincs minden [-jelnek]-párja	1 pont

5. feladat (8 pont)

A:	SZÓ = KABALA	3 pont
B:	KULCS = C	1 pont
C1:	A művelet KÓD(I-1) operandusából és KÓD(I) eredményéből	2 pont
	kiszámítható SZÓ(I) operandusa, és így a szó hátulról visszafelé haladva helyreállítható,	4 pont
vagy		
C2:	az itt alkalmazott XOR művelet asszociatív és kommutatív,	

ha két operandusa azonos, akkor 0 az eredménye; ha egyik operandusa 0, akkor az eredménye azonos a másik operandusával. Ezért $SZÖ(l) = KÖD(l)$ művelet $KÖD(l-1)$, ami az első betű kivételével mindig igaz. Az első betűre pedig $KULCS = KÖD(0) = SZÖ(1)$ művelet $KÖD(1)$ igaz.

6. feladat (16 pont)

- A: Csak a (*2*) változat kever szabályosan. (6 pont a hibátlan válaszra. Ebből 4 pont a levonás, ha a (*4*) változatot is szabályosnak vette; és tévedésenként 2 pont a levonás, ha az (*1*), (*2*) és (*3*) változatokra vonatkozó valamelyik válasz hibás.) max. 6 pont
- B: Az (*1*) és a (*3*) változat egyes lapokat megtöbbszörözhet, másokat kihagyhat. Az (*1*) változat nem jegyzi meg, hogy mely lapokat osztotta már ki. A (*3*) változat ezt ugyan megteszi, de a már egyszer kiosztott lapokból is újra oszt 1 pont
- A (*4*) változat minden lapot csak egyszer oszt ki, de nem szabályosan, mivel nem gondoskodik a lap helybenhagyásáról. Ezért például az első lap nem maradhat az első helyen. 4 pont
- C: A (*2*) változat a VÉLETLEN() függvényt a pakli leosztása közben várhatóan egyre gyakrabban hívja meg egy-egy lap kiosztásához, ugyanis egyre valószínűbb, hogy a függvény egy már kiosztott lap számát állítja elő 2 pont
- (Háttérinformáció a javító tanárnak: Az n-edik lap kiosztásához szükséges hívások száma geometriai eloszlást követ, amelynek várható értéke a kívánt esemény valószínűségének reciproka, esetünkben $32/(33-n)$. Ezért a teljes program lefutásához várhatóan $32/32 + 32/31 + 32/30 + \dots + 32/1$ hívás szükséges. Megsejtése plusz pont!) 1 pont

7. feladat (7 pont)

Nem csökkenő sorrendbe rendezve kiír 1 és 300 közötti egész számokat.

- Rendezve kiír: 3 pont
- növekvően: 2 pont
- 1 és 300 közötti egészeket: 2 pont

8. feladat (14 pont)

- A1: SELECT név, fizu 2 pont
FROM alkalmazottak 1 pont
WHERE munkakör = "osztvez" AND fizu > 10000 2 pont
- A2: SELECT név, osztnév 2 pont
FROM alkalmazottak, osztályok 2 pont
WHERE oszt = kód AND hely = "Pécs" 2 pont
- B: Kiírt nevek: Éber, Élő, Buzgó 2 pont
Minden hibáért egy pont levonás jár a 3-ból! 3 pont

+1 feladat

Nincs értékelési javaslat.

1990. Második fordulóElső-második osztályosok: Metróállomás

Értékelési szempontok:

Összesen:
100 pont**A. feladat:**

		15 pont
1.	érkezés	1 pont
	irányválasztás	1 pont
2.	sorba beállás	2 pont
3.	metrók érkezése TE időközönként	2 pont
	metrók várakozása TA időegységig	1 pont
4.	utasok kiszállása	1 pont
5.	várakozók beszállása	1 pont
6.	kiszállók kilépése az állomásról	1 pont
Eredmény:		2 pont
	két várakozó sor,	
	metró érkezése, indulása,	2 pont
	idő kijelzése	1 pont

B. feladat:

		15 pont
7.	sorbanállás	1 pont
8.	lépcsőrelépés	2 pont
9.	lefelé menő lépcső kezelése	2 pont
	felfelé menő lépcső kezelése	2 pont
10.	felfelé várakozók sora	1 pont
11.	(azonos a 6. részfeladattal)	0 pont
Eredmény:		2 pont
	két újabb sor,	
	lépcsők állapota,	4 pont
	lépcsők iránya	1 pont

C. feladat:

		15 pont
12.	váróterem megtelik	1 pont
13.	érkező metró az utasok száma	1 pont
	metrószerelevény betelik	1 pont
14.	szimuláció leáll, ha kell	1 pont
15.	egyszerre max. SZ számú utas	3 pont
	beszállók a kiszállók után	2 pont
16.	akik nem férnek el, nem szállnak ki	1 pont
Eredmény:		2 pont
	metró megtelt,	
	egyéb megtelések	3 pont

D. feladat:

		20 pont
17.	felhasználó lépcsőt indít	2 pont
	felhasználó lépcsőt leállít	2 pont

	ügyes lépcsőindítási stratégia	4 pont
	a harmadik lépcső mozgatása	2 pont
	a harmadik lépcsőre fellépés	2 pont
18.	automatikus indítás lefelé	2 pont
	automatikus indítás felfelé	2 pont
19.	peron megtelése (15. benne van)	1 pont
Eredmény:		2 pont
	harmadik lépcső kijelzése	
	harmadik lépcső iránya	1 pont
Programminőség (struktúra):		25 pont
	Tagolás soronként	3 pont
	Bekezdések	4 pont
	Magyarázatok	5 pont
	Strukturált szerkezetek	5 pont
	Eljárások alkalmazása	5 pont
	Beszédes azonosítók	3 pont
Programminőség (megjelenítés):		10 pont
	Programparaméterek (bemeneti adatok) – mit, hogyan	3 pont
	Programaraméterek ellenőrzése	2 pont
	Képernyő esztétikussága	5 pont
Megjegyzések:		
-	az egyes részfeladatokra pluszpontok adhatók,	
-	programparaméterek helyett használhatók alkalmasan megválasztott állandók.	

Harmadik-ötödik osztályosok: Szóelválasztás

Értékelési útmutató

(Tesztelésre csak értelmes szavakat érdemes használni, mert a programoknak a magyar nyelv szabályait nem kell tesztelnie, és hibás szavakat sem kell felismernie.)

Összesen:
105 pont

A) Általában:

	Egyértelmű esetet sohasem tekint kétesnek.	20 pont
	Billentyűzetről működik.	2 pont
	Állományból működik.	3 pont
	Állomány végét felismeri.	1 pont
	Szóközzel tagolható a szöveg.	1 pont
	Sorvéggel tagolható a szöveg.	1 pont
	Egyenlőségjelet felismeri.	1 pont
	A kétes eseteket egyenként végigkérdezi, és végül a teljes szöveget egyben írja ki.	5 pont
	Ékezetes magánhangzókat felismeri.	2 pont
	Nagy- és kisbetűket jól kezeli.	2 pont

B) Egyszerűbb elválasztások (1.-8.szabályok):

	Egyszótagút nem választ el (pl. bal).	20 pont
	Több szótagút elválaszt, ha a két magánhangzó között	1 pont
-	egyjegyű rövid mássalhangzó van (pl. ka-lap);	
-	több egyjegyű rövid mássalhangzó van (pl. tor-ta);	1 pont
-	lehet egyjegyű hosszú mássalhangzó is (pl. dur-ran, elv-vel, csekk-hez);	1 pont

-	többjegyű rövid mássalhangzó egyedül áll (pl. bo-dza);	2 pont
-	többjegyű rövid mássalhangzó nem egyedül áll (pl. mor-zsa, fin-dzsa);	2 pont
-	többjegyű hosszú mássalhangzó egyedül áll (pl. meny-nyi, bridzs-dzsel);	2 pont
-	többjegyű hosszú mássalhangzó nem egyedül áll (pl. menny-be, kulcs-csal).	1 pont
	X-et jól választja el (pl. pra-xis).	1 pont
	Ch-t többjegyűnek veszi (pl. or-chi-dea).	1 pont
	X-val,-vel,-vá,-vé ragos alakját jól választja el (pl. bórax-szá).	2 pont
	Ch-val,-vel,-vá,-vé ragos alakját jól választja el (pl. pech-hel).	2 pont
	Kétes esetben többet is kijelez (pl. srác-hoz, srá-choz).	3 pont
C) Bonyolultabb elválasztások (9.-16. szabályok):		30 pont
	Kötőjellel írt összetett szót jól választ el, azaz nem tesz ki két kötőjelet (pl. tip-top).	1 pont
	Egyenlőségjelet (=) többjegyű betű esetén és két magánhangzó között jól értelmez (pl. ház-szem-le, cse-re-a-lap).	1 pont
	Jól felismeri és leválasztja a leg-et (pl. leg-a-lább).	1 pont
	Jól felismeri és leválasztja az igekötőket (pl. fel-e-sel, ki-próbál, meg-int). 2 pont, ha csak néhányat ismer fel.	4 pont
	Jól felismeri és – csak szó végéről – leválasztja a -ság,-ség,-szerű stb. képzőket száz-szor (pl. ma-lac-ság, száz-szor, ész-sze-rű). 2 pont, ha csak néhányat ismer fel.	4 pont
	Leg-, igekötő, -ság,-ség,-szerű stb. esetén kétes esetben több változatot kijelez (pl. le-ga-lább, leg-a-lább, fel-e-sel, fe-le-sel, meg-int, me-gint, ma-lac-ság, ma-la-cság, száz-szor, százs-zor), ui. ha az értelmét nem ismerjük, nem tudhatjuk, melyik a helyes.	3 pont
	Igekötőknél, -ság,-szőr stb. esetén ékezetes és ékezet nélküli magánhangzók között különbséget tesz.	1 pont
	Két magánhangzó közé nem tesz kötőjelet (pl. tea-fű).	2 pont
	Mégis tesz két magánhangzó közé kötőjelet, ha ott magánhangzóra végződő igekötő van (pl. le-áll, ki-e-mel). 1 pont ha csak néhányra tudja.	2 pont
	Ilyenkor mindkét esetet felkínálja, mert nem tudhatja, melyik a jó (pl. le-áll, leáll, ki-e-mel, kie-mel).	2 pont
	Igekötő, ill. leg- után, ha nem teljes a szótag, akkor ott nem választ el (pl. kü-lönb, be-lem, fe-le).	3 pont
	Szó elején álló magánhangzót nem választ le (pl. idill).	2 pont
	Szó végén álló magánhangzót nem választ le (pl. ké-mia).	2 pont
	Szó elején álló magánhangzó-csoportot nem választ le (pl. Euró-pa).	2 pont
Programminőség (struktúra):		25 pont
	Tagolás soronként	3 pont
	Bekezdések	4 pont
	Magyarázatok	5 pont
	Strukturált szerkezetek	5 pont
	Eljárások alkalmazása	5 pont
	Beszédes azonosítók	3 pont
Programminőség (megjelenítés):		10 pont
	Kényelmes adatbevitel	5 pont
	Képernyő esztétikussága	5 pont

1991. Első fordulóElső–második osztályosok

Összpontszám: 56 pont

Beküldési
határ: 23 pont

Postázási határidő: 1991. január 31.

1. feladat (5 pont)

A, B, D: "ROSSZI"; C, E: "JÓ". Minden helyes válaszra

1-1 pont

2. feladat (8 pont)

A. a. testvér(x,y). Jobb: testvér(x,y) VAGY féltestvér(x,y).

2 pont

b. nagyapja(x,y).

2 pont

B. a. anyaiNagyszülője(x,y) HA szülője(x,z) ÉS anyja(z,y).

2 pont

b. szülőpár(x,y) HA szülője(x,z) ÉS szülője(y,z) ÉS $x \diamond y$.

2 pont

3. feladat (8 pont)

A. Helyes.

1 pont

B. Hibás, mert Evő megpróbálhat palacsintát felvenni a tányérról, mielőtt Sütő szólna, hogy EHETSZ.

2 pont

C. Hibás, mert ha Sütő elég gyors, akkor az ÜRES A TÁNYÉR jelzésre hamarabb ráteheti a következő palacsintát a tányérra, minthogy Evő elvenné az ott levőt.

2 pont

D. Helyes.

1 pont

D lassúbb A-nál, mert Sütő csak azután folytathatja a sütést, miután Evő megette a palacsintát, azaz amíg Evő eszik, Sütő nem rakhatja a tányérra a következőt.

2 pont

4. feladat (12 pont)

A: Bármilyen szöveg, amelyben LILILLA vagy LILLILLA előfordul. Semmilyen más megoldás nem jó.

2-2 pont

B: 40 IF K\$ <> "L" THEN IR("LI"): GOTO 20 ELSE OLVAS(K\$)

4-4 pont

50 IF K\$ <> "A" THEN IR("LIL"): GOTO 20 ELSE GOTO 10

Más megoldások is elfogadhatók, de a fentieknél bonyolultabbakra csak 2-2 pont adható.

5. feladat (11 pont)

A: a, d: igen; b, c: nem.

Két vagy három helyes válasz esetén:

1 pont

ill. ha mind a négy válasz helyes:

2 pont

- B: Nincs igaza. A gép 3-mal osztható összegeket hajlandó kifizetni. (Ui. a piros lámpa akkor gyullad fel, ha az addig bedobott összeg osztható 3-mal. A zöld lámpa ég, ha 1 a maradék, és a kék ég, ha 2 a maradék.) 4 pont
- C: 510 Ft-ot. (Ui. $2^9 - 1 = 511$, de mivel ez nem osztható 3-mal, 510 a helyes megoldás.) 2 pont
- D: Legalább 6 zsetont. (Ui. 6 zsetonért 60 Ft-ot kell fizetnem, és 63 Ft-ot nyerhetek vele.) 3 pont

6. feladat (5 pont)

- A. 9 2 pont
- B. $2 - 2 = 0$ 1 pont
- C. $Y - X$ 2 pont

7. feladat (7 pont)

- Székesfehérvár 1 pont
- Veszprém, Fonyód, Dombóvár 3 pont
- Szombathely 1 pont
- Nagykanizsa, Kaposvár 2 pont
- de ha Kaposvárt nem adja meg 0 pont
- ha Pécs is szerepel a listán -2 pont

Azaz városonként 1-1 pont, amíg a sorrend megegyezik az itt megadottal.

Harmadik-ötödik osztályosok

Összpontszám: 94 pont

Beküldési
határ: 38 pont

Postázási határidő: 1991. január 31.

1. feladat (12 pont)

- A. A visszatérési értékek, ill. a tömb tartalma a két függvényeljárás esetén:
nyissz: 1, 2, 3, 4 és $[0, 1, 0, 2, 7, -1]$ 3 pont
nyassz: 1, 4, 3, 2 és $[0, 3, 0, 0, 7, -1]$ 3 pont
- B. 1, 2 esetén nyissz a 2-re ad 0-t eredményül, nyassz jó. 2 pont
- C. 1, 2, 2 esetén nyassz az utolsó 2-esre 0-t ad eredményül, nyissz jó. (A B. és C. kérdésekre nincs más jó megoldás!) 4 pont

2. feladat (10 pont)

- A a. testvér(x,y). Jobb: testvér(x,y) VAGY féltestvér(x,y). 1 pont
b. nagyapja(x,y). 1 pont
c. őse(x,y). 2 pont
- B. a. anyaiNagyszülője(x,y) HA szülője(x,z) ÉS anyja(z,y). 1 pont
b. szülőpár(x,y) HA szülője(x,z) ÉS szülője(y,z) ÉS $x < y$. 2 pont
c. leszármazottja(x,y) HA szülője(y,x) VAGY (szülője(y,z) ÉS leszármazottja(x,z)). 3 pont

3. feladat (8 pont)

- A. Helyes. 1 pont
- B. Hibás, mert Evő megpróbálhat palacsintát felvenni a tányérról, mielőtt Sütő szólna, hogy EHETSZ. 2 pont
- C. Hibás, mert ha Sütő elég gyors, akkor az ÜRES A TÁNYÉR jelzésre hamarabb ráteheti a következő palacsintát a tányérra, minthogy Evő elvinné az ott levőt. 2 pont
- D. Helyes. 1 pont
D lassúbb A-nál, mert Sütő csak azután folytathatja a sütést, miután Evő megette a palacsintát, azaz amíg Evő eszik, Sütő nem sütheti a következőt. 2 pont

4. feladat (15 pont)

A bal oldali kiválasztó, a jobb oldali pedig buborékrendezés. Mindkét algoritmus nagyság szerint (nem szigorúan monoton) növekvő sorba rendez.

- A. állítás: Minden h -ra 1 és $l-2$ között: $A(h) \leq A(h+1)$, és minden k -ra l és N között: $A(l-1) \leq A(k)$. 2 pont
Szóban: Az $A(1..l-1)$ résztömb elemei nagyság szerint (nem szigorúan monoton) növekvő sorrendben vannak, és közülük a legnagyobb – $A(l-1)$ – sem nagyobb az $A(l..N)$ résztömb bármely eleménél. 2 pont
- B. állítás: Teljesül az A. állítás, és minden k -ra l és $J-1$ között: $A(M) \leq A(k)$. 2 pont
Szóban: Teljesül az A. állítás, és az $A(M)$ elem biztosan nem nagyobb az $A(l..J-1)$ résztömb bármely eleménél. 2 pont
- C. állítás: Minden h -ra 1 és $l-2$ között: $A(h) \leq A(h+1)$. 2 pont
Szóban: Az $A(1..l-1)$ résztömb elemei nagyság szerint (nem szigorúan monoton) növekvő sorrendben vannak.
- D. állítás: Minden h -ra 1 és $J-1$ között: $A(h) \leq A(h+1)$, és $A(J) > A(J+1)$, és minden k -ra $J+1$ és $l-1$ között: $A(k) \leq A(k+1)$. 2 pont
Szóban: Az $A(1..J)$ résztömb elemei nagyság szerint (nem szigorúan monoton) növekvő sorrendben vannak, és $A(J)$ nagyobb $A(J+1)$ -nél, és az $A(J+1..l)$ résztömb elemei is nagyság szerint (nem szigorúan monoton) növekvő sorrendben vannak.

5. feladat (12 pont)

- A. $-1, 26, 2, 117, 8, 18, -1$ 5 pont
- B. $18, -1, 2, -1, 13, 14, -1, -1, 26, 117, 8$ 7 pont
- Azaz minden jó helyre írt jó érték után 1-1 pont jár.

6. feladat (12 pont)

- Hibátlan átvitel esetén i értéke 0. 1 pont
- Egyszeres hiba esetén i 1 és 7 közötti érték, a meghibásodott bit sorszáma. 2 pont

3, 5, 7,
3, 6, 7,
5, 6, 7.

Minden
helyes szám;
1 pont

7. feladat (12 pont)

- A: Bármilyen szöveg, amelyben LILILLA vagy LILLILLA előfordul. Semmilyen más megoldás nem jó. 2-2 pont
- B: 40 IF K\$ <> "L" THEN IR("LI"): GOTO 20 ELSE OLVAS(K\$) 4 pont
50 IF K\$ <> "A" THEN IR("LIL"): GOTO 20 ELSE GOTO 10 4 pont
Más megoldások is elfogadhatók, de a fentieknél bonyolultabbakra csak 2-2 pont adható.

8. feladat (13 pont)

- A. A d változóban az aktuális haladási irányt tárolja a program: 1: jobbra, 2: le, 3: balra, 4: fel. 1 pont
(Ha csak egy válasz is rossz, nem jár érte a pont!)
B. A jó megoldásokra egyenként 3-3 pont adható. Kis hiba esetén (pl. egy-két számot hibásan írt be, de egyébként következetes a megoldás) 1-1 pontot le kell vonni. Ha a ** -gal megjelölt mezőkbe beírja a következő sorszámot, akkor ugyancsak le kell vonni 1-1 pontot, ugyanis az eljárás az utolsó mezőt sohasem tölti. 4*3 pont

a)

	1	2	3	4	
15	16	17	18	5	6
14	13	**	19	8	7
	12	11	10	9	

b)

	5	4	3	**	
	6	1	2	11	
	7	8	9	10	

c)

		5	4	3	
		6	1	2	
		7	8	**	

d)

	9	10	11	12	
7	8	19	**	13	14
6	5	18	17	16	15
	4	3	2	1	

1991. Második fordulóElső-második osztályosok: 1. Hetedhét

Értékelési szempontok:

Összesen: 40
pont

Tesztesetek a Szűkszámokhoz:

A	1	jól kezeli a számjegyeket	
E	7		
O	49		
K	343		
T	2401		
H	16807		
AAA	3	jól kezel egymás mellett 3 számjegyet	
EO	42	jól kezel előtagot	
KKKT	1372	jól kezel 3 előtagot	
KO	392	jól kezel utótagot	
TKKK	3430	jól kezel 3 utótagot	
EOA	43	jól kezel elő- és utótagot egyszerre	
AOAE	54	előtag lehet több helyen	
AAT	2399	szóközt szűr	
M	117649		
MHTKOE A	137257		
A.	Beolvasás		3 pont
B.	Kiírás		3 pont
	csak képernyőre:	1 pont	7 pont
	csak állományba:	2 pont	
C.	Összes szűkszámjegy felismerése (PC-n: Hosszú egésszek kezelése is) számjegyenként:	1-1 pont	7 pont
D.	Előtagok kivonása előtagok felismerése:	3 pont	7 pont
	mind a 3 előtag felismerése:	2 pont	
	jó érték levonása:	2 pont	
E.	Utótagok hozzáadása utótagok felismerése:	3 pont	3 pont
	mind a 3 utótag felismerése:	2 pont	
	jó érték hozzáadása:	2 pont	
F.	Szóközők szűrése		
G.	Programszerkezet, stílus		10 pont
	szabályos programszerkezetek	2 pont	
	megfelelő sorokra tagolás	1 pont	
	bekezdéses leírás	2 pont	
	sorok közötti tagolás	1 pont	
	eljárások alkalmazása	2 pont	
	magyarázatok	1 pont	
	beszédes azonosítók	1 pont	

Első–második osztályosok: 2. Útkeresztveződés

Értékelési szempontok:

	Összesen: 100 pont
A. Ábrázolás	
Az útkeresztveződés kirajzolása	14 pont
Paraméterek beolvasása (sávonként 1)	6 pont
Darabszámok kijelzése (sávonként 1)	4 pont
B. Autók mozgása	
Az autók vízszintesen mozognak	26 pont
A keresztveződésnél tudnak balra fordulni	3 pont
A keresztveződésnél tudnak jobbra fordulni	3 pont
Az autók függőlegesen lefelé mozognak	3 pont
Az autók függőlegesen felfelé mozognak	3 pont
A keresztveződésnél fordulni is tudnak	3 pont
Vízszintesen véletlenszerűen belépnek	3 pont
Függőlegesen véletlenszerűen belépnek	3 pont
A vízszintes út végén eltűnnek	1 pont
A függőleges út végén eltűnnek	1 pont
C. Lámpakezelés	
Négyféle lámpát kezel (piros, pirossárga, zöld, sárga)	20 pont
A lámpaváltások időzítése (a sárga rövid)	4 pont
Jól kezeli az ellentétes irányú lámpát	4 pont
A forgalomtól függően állítja a lámpát:	
van min-max idő a zöld lámpára,	2 pont
az úton lévő aktuális számától függ,	4 pont
darabszámmal arányos	4 pont
D. Autók megállása, elindulása	
Az autók piros, sárga lámpánál megállnak	15 pont
Álló autó mögött a következők megállnak	5 pont
Álló autó elindul, ha lehetséges	5 pont
E. Programszerkezet	
Szabályos programszerkezetek	25 pont
Megfelelő sorokra tagolás	5 pont
Bekezdéses leírás	3 pont
Sorok közötti tagolás	4 pont
Eljárások alkalmazása	2 pont
Magyarázatok	5 pont
Beszédes azonosítók	3 pont

Harmadik–ötödik osztályosok: Szó ami szó

Értékelési szempontok:

Összesen:
140 pont

Tesztszöveg a Szó ami szóhoz:

2. A jelzolampak pillanatnyi állapotának megjelenítése.
3. Az utakon mozgo, varakozo autok szamanak kijelzese.
4. Az autok a ket utra veletlenszeruen érkeznek, ez legyen parameterezhető, az uton a vegeig haladnak, ameddig a képernyon abra-
zolja az utat.
7. A lampavaltasok idejet a felhasznalo adja meg, amelyet a program-
nak a forgalomtól függően modositania kell.
8. Az autok vagy állnak, vagy azonos sebesseggel haladnak, egy-
mast nem előzhetik.

minta:

a bekezdés elé írt sorszám:

autok	3, 4, 8	
*lampak	2	
ut*	3, 4	
mo*o	3	
lam*va*ok	7	
ss	8	
mozgo & autok & varakozo	3	
harom + ket + egy	4	
lathato & autok + allnak & haladnak	8	
!autok	2, 7	
jelzolampak & !autok		2
!autok & jelzolampak		2
jelzolampak + (mozgo + !autok) & adja	2, 7	
semmi + !(autok + akarmi)	2, 7	

A. Szövegbeolvasás

5 pont

B. Mintabeolvasás

- billentyűzetről
- fájlból
- választani lehet

5 pont

1 pont

3 pont

1 pont

C. Kiírás

- képernyőre
- állományba

8 pont

3 pont

5 pont

D. Bekezdéskezelés

7 pont

E. Szóillesztés

- szódarabolás, határoló karakterek felismerése
- illesztés joker nélkül
- illesztés egy jokerrel
- illesztés több jokerrel

25 pont

2 pont

5 pont

10 pont

8 pont

F. Kifejezésértékelés

- ÉS
- VAGY
- TAGADÁS
- precedencia
- zárójeles kifejezés

60 pont

10 pont

10 pont

10 pont

5 pont

25 pont

G. Programszerkezet, stílus

Szabályos programszerkezetek	30 pont
Megfelelő sorokra tagolás	6 pont
Bekezdéses leírás	4 pont
Sorok közötti tagolás	5 pont
Eljárások alkalmazása	3 pont
Magyarázatok	6 pont
Beszédes azonosítók	3 pont
	3 pont

1992. Első fordulóElső-második osztályosok

Összpontszám: 100 pont

Beküldési
határ: 40 pont

Postázási határidő: 1992. január 31.

1. feladat (10 pont)

- | | | |
|----|--------------------------------------|--------|
| A. | 5 | 1 pont |
| | 4 | 1 pont |
| | 6 | 2 pont |
| B. | A két szám legnagyobb közös osztója. | 6 pont |

2. feladat (15 pont)

- | | | |
|-----|---|----------------------------|
| A1. | "Ipafai fapipa" | 1 pont |
| A2. | "Ézüsttel befuttatott." | 1 pont |
| A3. | "13"+CHR\$(0)+CHR\$(7)+"-szor ismételd meg." | 1 pont |
| A4. | "Gyakorisága 8" | 1 pont |
| B. | Szöveget tömörít,
a 2-nél többször ismétlődő karaktereket
a karakter + CHR\$(0) + CHR\$(darabszám) hármassal
helyettesíti. | 2 pont
1 pont
1 pont |
| C. | Ha A\$ két vagy több azonos karakterrel ér véget,
ha 255-nél több egyforma karakter van A\$-ban. | 3 pont
2 pont |
| D. | Az 1100-as és 1110-es sorba át kell másolni az 1040-es,
ill. az 1050-es sorok tartalmát. (Bármilyen, ezzel azonos
hatású megoldás is jó.) | 1-1 pont |

3. feladat (13 pont)

- | | | |
|----|---|--------|
| A. | B(j) az A vektor értékű elemeinek száma: | 3 pont |
| B. | B(j) az A vektor értékű vagy j-nél kisebb elemeinek száma: | 3 pont |
| C. | B(j) az A vektor j-nél kisebb elemeinek száma: | 3 pont |
| | C-ben az A vektor elemei vannak nagyság szerint növekvő sorrendben: | 4 pont |

4. feladat (18 pont)

- F1. Egy lehetséges megoldás:

F1(I, T, J, K):

K:=1

Ciklus amíg K \uparrow T és I + K \uparrow N és A(I+K,J) = -1

K := K + 1

Ciklus vége

Ha I + K \uparrow N akkor K := K - 1

Eljárás vége.

6 pont

Részpontszámok:

K 0 és T közötti érték: 2 pont

autót nem lép át: 2 pont

I + K = N + 1 eset jó: 2 pont

F2. A(I+K,J) := A(I,J) + 1 : A(I,J) := -1

3 pont

F3. A(I+K,J) := A(I,J) : A(I,J) := -1

3 pont

F4. Lassítás

3 pont

F5. Sáv váltás

3 pont

5. feladat (8 pont)

A. Mitsubishi autóembléma: azaz

3 db

1 pont

x élhosszúságú

1 pont

rombusz.

1 pont

120-120 fokkal elforgatva)

1 pont

B. Audi autóembléma, azaz

4 db

1 pont

10 egységnyire egymásba fonódó

1 pont

360/(2*pi) sugarú

1 pont

kör.

1 pont

6. feladat (13 pont)

A. '1' - ettől kezdve egy szakasz végpontja mozog a képen,

2 pont

(J, B, L, F hatására) kezdőpontja az ekkori kurzorhelyzet.

2 pont

B. '2' - ettől kezdve egy téglalap egyik sarka mozog a képen,

3 pont

szembenlevő sarka az ekkori kurzorhelyzet.

3 pont

C. Az egyes funkciók (1 és 2) által rajzolt alakzatokat egy tetszőleges újabb funkció (1, 2) kiválasztása zárja le.

3 pont

7. feladat (10 pont)

E1. Két szín keverékszíne (S3-ban):

3 pont

E2. Egy szín kiegészítő színe (S2-ben azok az alapszínek vannak, amelyek S1-ben nincsenek):

3 pont

E3. Egy szín (S1) felbontása alapszínekre (SZ):

3 pont

I-ben az előforduló alapszínek száma:

1 pont

8. feladat (13 pont)

- | | | |
|----|---|------------------|
| A. | Az L2 karaktersorozat az L1-hez fűzi: | 4 pont |
| B. | Az L karaktersorozatból kitörli az E elemet:
de csak E első előfordulását vizsgálja: | 3 pont
2 pont |
| C. | "NEMES+TIHAMER": | 2 pont |
| D. | "NMES": | 2 pont |

Harmadik-ötödik osztályosok

Összpontszám: 120 pont

Beküldés
határ: 48 pont

Postázási határidő: 1992. január 31.

1. feladat (18 pont)

- | | | |
|----|--|------------------|
| A. | -1, -2, -3, -4: az összeadás, kivonás, szorzás, osztás
azonosítói,
pozitív számok: aritmetikai műveletek operandusai. | 2 pont
2 pont |
| B. | $7 - 5 = 2$ | 2 pont |
| C. | $2 * 2 + 6 / 2 = 7$ (jó műveletenként 2-2 pont) | 6 pont |
| D. | 0-vel osztásakor,
ismeretlen művelet végrehajtásakor (azaz ha nem -1,
-2, -3 vagy -4 a művelet kódja),
ha éppen -1 a végeredmény. | 2 pont
2 pont |

2. feladat (15 pont)

- | | | |
|-----|---|----------------------------|
| A1. | "Ipafai fapipa" | 1 pont |
| A2. | "Ezüsttel befuttatott." | 1 pont |
| A3. | "13"+CHR\$(0)+CHR\$(7)+"-szor ismételd meg." | 1 pont |
| A4. | "Gyakorisága 8" | 1 pont |
| B. | Szöveget tömörít,
a 2-nél többször ismétlődő karaktereket
a karakter + CHR\$(0) + CHR\$(darabszám) hármassal
helyettesíti. | 2 pont
1 pont
1 pont |
| C. | Ha A\$ két vagy több azonos karakterrel ér véget,
ha 255-nél több egyforma karakter van A\$-ban. | 3 pont
2 pont |
| D. | Az 1100-as és 1110-es sorba át kell másolni az 1040-es,
ill. az 1050-es sorok tartalmát. (Bármilyen, ezzel azonos
hatású megoldás is jó.) | 1-1 pont |

3. feladat (13 pont)

- | | | |
|----|--|------------------|
| A. | B(j) az A vektor értékű elemeinek száma: | 3 pont |
| B. | B(j) az A vektor értékű vagy j-nél kisebb elemeinek száma: | 3 pont |
| C. | B(j) az A vektor j-nél kisebb elemeinek száma:
C-ben az A vektor elemei vannak nagyság szerint nö-
vekvő sorrendben: | 3 pont
4 pont |

4. feladat (18 pont)

F1. Egy lehetséges megoldás:

F1(I, T, J, K):

K:=1

Ciklus amíg $K \uparrow T$ és $I + K \uparrow N$ és $A(I+K, J) = -1$

K := K + 1

Ciklus vége

Ha $I + K \uparrow N$ akkor K := K - 1

Eljárás vége.

6 pont

Részpontszámok:

K 0 és T közötti érték: 2 pont

autót nem lép át: 2 pont

 $I + K = N + 1$ eset jó: 2 pontF2. $A(I+K, J) := A(I, J) + 1$; $A(I, J) := -1$ 3 pontF3. $A(I+K, J) := A(I, J)$; $A(I, J) := -1$ 3 pont

F4. Lassítás 3 pont

F5. Sávváltás 3 pont

5. feladat (14 pont):

A. Az (I, J) és (I, J+1) közötti metszéspont koordinátáinak kiszámolása (M1, M2 és S2 új változók): 8 pont

 $M1 := \text{abs}(T(I, J) - T(I, J+1))$ $M2 := \text{abs}(T(I, J) - SZ)$ $S2 := S * M2/M1$ metszéspont koordinátái: $I^*S - S, J^*S - S + S2$ Részpontszámok: aránypár $[M2:M1 = S2:S]$: 2 pont,

M1: 2 pont, M2: 1 pont, S2: 1 pont, metszéspont: 2 pont.

B. Egy pontnégyesen belül:

nincs metszéspont (mind a 4 pont vagy SZ alatt, vagy SZ felett van): 2 pont

két metszéspont van (egy van SZ alatt és három SZ felett; vagy egy van SZ felett és 3 SZ alatt; vagy 2 szomszédos pont van alatta és a másik 2 felette): 3 pont

négy metszéspont van (2 szemközti pont van alatta és a másik 2 felette): 1 pont

elfajuló esetek nincsenek, mert színtvonal rácspont fölött

nem haladhat át. Ha mégis megenged 1 vagy 3 metszéspontot, 2 pont levonandó.

Megjegyzés: A feladat megfogalmazása és pontozása sokat változott az 1992-es feladatsorhoz képest. Az eredeti szöveg többek között az algoritmus átírását is kérte, mégpedig úgy, hogy a színtvonal az A) pont szerinti metszéspontokat kösse össze egyenes szakaszokkal, és ne menjen át a pontnégyesek középpontján. Mivel a lehetséges esetek megvizsgálásához viszonylag hosszú programot kellett volna írni, és erre a versenyzőknek nem volt idejük, ezt a feladatrészt elhagytuk.

6. feladat (16 pont)

A. Bértéke 1,2 vagy 3 lehet.

1-1-1 pont

- B. B-t a 12330-as sorban módosított A változó értékének utolsó számjegyével növeli meg. 2 pont
A ennek az értéknek a tizedrésze (az utolsó számjegyet hagyja el B-ből). 2 pont
- C. Legfeljebb 3 kódot kell kipróbálnia. 1 pont
Ki kell próbálnia az alábbi kódokat:
B = 3 esetére a 183-at, és 1 pont
B = 2 esetére a 133-at vagy a 188-at, és 1-1 pont
B = 1 esetére a 138-at vagy a 184-et vagy a 193-at 1-1-1 pont
ha B=2 és B=1 esetén utal a választhatóságra 1-1 pont

7. feladat (13 pont)

- A. A függvény \sqrt{x} közelítő értékét adja meg. 5 pont
B. M a pontosságot írja elő. 2 pont
C. $A \leq \sqrt{x} \leq B$, az $[A, B]$ intervallum felezéssel egyre csökken. 3 pont
D. $A = 0$, 1 pont
 $B = X + 1$ 2 pont
Magyarázat: $A \leq \sqrt{x} \leq B$, ezért
- ha $X > 1$, akkor $0 < \sqrt{x} < X < X+1$
- ha $0 \leq X \leq 1$, akkor $0 \leq \sqrt{x} \leq 1 \leq X+1$

8. feladat (13 pont)

- A. Hatása: ismerjfel egy új listát ad eredményül azzal, hogy l2-t l1-hez fűzi – ezért a neve pl. fűzzHozzá lehetne. 2 pont
Működése: ha l1 nem üres, először is leválasztja róla az elemet (a fejét), majd feltételezve, hogy már létezik az l1 farkából és az l2-ből (saját maga rekurzív meghívásával) összefűzött lista, hozzálétre segítségével létrehozza a kívánt listát. 3 pont
- B. Hatása: találjki egy új listát ad eredményül azzal, hogy az első e elemet kitörli az l listából; ha nincs benne, akkor magát l-et adja vissza – ezért a neve pl. töröjki lehetne. 2 pont
Működése: ha l nem üres, megnézi, hogy a feje e-vel egyenlő-e; ha igen, e-t elhagyva l farka az eredmény; ellenkező esetben feltételezi, hogy már létezik az l farkából az e elem kitörésével kapott lista (saját maga rekurzív meghívásával), és így az l fejéből és az e-t már biztosan nem tartalmazó listából létrehozza az eredményt. 3 pont
- C. $N(E(M(E(S(+T(I(H(A(M(E(R)))))))))))))$ 1 pont
D. $(N(M(E(S))))$ 2 pont

1992. Második fordulóElső–második osztályosok: Demográfia

Értékelési szempontok:

Összesen:
100 pont

Adatbeolvasás

Kezdőállapot beolvasása	12 pont
Halálozási paraméterek beolvasása	3 pont
Születési paraméterek beolvasása	3 pont
Járványparaméterek beolvasása	3 pont

A beolvasások 3 pontjának megoszlása:

van: 1 pont, kérdés: 1 pont, ellenőrzés: 1 pont

A. Alapvető szimulációs feladatok

Halálozás jó megoldása	21 pont
Születés jó megoldása	7 pont
Öregedés jó megoldása	7 pont
	7 pont

B. Járvány

Járvány véletlenszerűen tör ki	12 pont
Járvány esetén mások a halálozási ráták	5 pont
	7 pont

C. Létszámtúllépés

Túllépés felismerése	10 pont
Túllépés megszüntetése	3 pont
Arányos megszüntetés	2 pont
	5 pont

Grafikon

Korcsoporteloszlás-grafikon van képernyőre/ablakra normált ez a grafikon skálabeosztás, feliratok vannak rajta	25 pont
Populációlétszám-grafikon van képernyőre/ablakra normált ez a grafikon skálabeosztás, feliratok vannak rajta	6 pont
időbeli betelés problémája meg van oldva	2 pont
	2 pont
	2 pont
	2 pont
	5 pont

Programminőség

Értelmesen sorokra tördelt a programszöveg	20 pont
Bekezdéses leírás a struktúrák elejének, végének jelzésével	2 pont
Programegységek tagolása (elválasztó sorokkal)	4 pont
Eljáráások alkalmazása	2 pont
Megjegyzések, magyarázatok	4 pont
Beszédes azonosítók	4 pont
Állandók (szinonimák) alkalmazása	2 pont
	2 pont

Harmadik-ötödik osztályosok: Petri-gráf

Értékelési szempontok:

Tesztelésre alkalmas hálózatok

Háló-1: 3 csomópont, 1 átmenet (holtpont egy lépésben)

"a" átmenetbe bemegy a nyíl: "1" csomópontból, ahol van bigyó

"a" átmenetbe bemegy a nyíl: "2" csomópontból, ahol nincs bigyó

"a" átmenetből kimegy a nyíl: "3" csomópontba

Háló-2: 3 csomópont, 1 átmenet (holtpont két lépésben)

"a" be: 1, 2, ki: 2, 3

Háló-3: 1 csomópont, 1 átmenet (ciklus)

"a" be: 1, ki: 1

Háló-4: 2 csomópont, 1 átmenet (ciklus lehetősége)

"a" be: 1, ki: 1, 2

Háló-5: 5 csomópont, 4 átmenet (ez a 3. példa hálózata)

"a" be: 1, ki: 2

"b" be: 1, ki: 3

"c" be: 2, ki: 1, 5

"d" be: 3, ki: 1, 4

Háló-6: 5 csomópont, 8 átmenet

"a" be: 1, ki: 2

"b" be: 2, ki: 1

"c" be: 2, ki: 3

"d" be: 3, ki: 1

"e" be: 3, ki: 4

"f" be: 4, ki: 1

"g" be: 4, ki: 5

"h" be: 5, ki: 1

Elérhető-e az 10000 állapotból a 00010 állapot? (igen)

Elérhető-e a 00001 állapotból az 10010 állapot? (nem)

Elérhető-e a 01110 állapotból a 00300 állapot? (igen)

Háló-7: 4 csomópont, 4 átmenet

"a" be: 1, 2, ki: 3

"b" be: 3, ki: 1, 2

"c" be: 3, ki: 4

"d" be: 4, ki: 3

Elérhető-e az 1100 állapotból a 0001 állapot? (igen)

Elérhető-e az 1100 állapotból a 0020 állapot? (nem)

Elérhető-e az 1111 állapotból a 3300 állapot? (igen)

Háló-8: 4 csomópont, 4 állapot

"a" be: 1, ki: 2, 3

"b" be: 1, 3, ki: 4

"c" be: 4, ki: 3

"d" be: 2, ki: 1

Elérhető-e az 1000 állapotból a 0010 állapot? (igen)

Elérhető-e a 0010 állapotból az 1000 állapot? (igen)

Elérhető-e az 1111 állapotból a 0002 állapot? (igen)

Háló-2 - Háló-8-ban feltesszük, hogy kezdetben minden bemenő csomópontban van bigyó.

1. feladat:

1.1. A hálózat szerkezete megadható, a csomópontokat számjegyek (1, 2, ...), az átmeneteket betűk (a, b, c, ...) jelölik:

1.2. A tárolt gráf kiírható:

1.3. Megadható a kezdőállapot (melyik csomópontban hány bigyó van):

1.4. Az előző kérdésekre nem lehet értelmetlen adatot megadni, pl. negatív csomópontszám, negatív bigyószám, nem létező helyre irányuló nyíl, független csomópont, átmenet bemenő nyíl nélkül, átmenet kimenő nyíl nélkül:

1.5. Nem lehet a feltételnek ellentmondó hálózatot megadni, azaz nem lehetnek párhuzamos nyilak és 1-nél több bigyó a kezdeti állapotban:

23 pont

4 pont

1 pont

1 pont

4 pont

2 pont

1 pont

1 pont

1 pont

1 pont

3 pont

2 pont

2. feladat:

- 2.1.A szimuláció működik,
az egyes állapotokat kijelzi: 27 pont
10 pont
5 pont
- 2.2.A versenyhelyzeteket kezeli, azaz az egyidejűleg aktivizálható át-
menetek közül véletlenszerűen választ: 5 pont
- 2.3.Végtelen működési sorozat esetén a 8. állapot után a szimuláció le-
áll: 2 pont
- 2.4.A holtpontot – a 8. lépés megtétele előtt már nincs aktivizálható át-
menet – észreveszi és kijelzi: 5 pont

3. feladat:

- 3.1.A program az adott hálózathoz és kezdőállapothoz előállítja és ki-
írja a 8 lépésben elérhető állapotok halmazát: 30 pont
15 pont
- 3.2.Észreveszi, ha egy már korábban bejárt állapotba jut (ciklus-1): 5 pont
- 3.3.Észreveszi, ha olyan állapotba jut, amely egy előzőleg már figye-
lembevett állapottól csak abban különbözik, hogy egyik csomó-
pontban sincs kevesebb bigyó (ciklus-2): 5 pont
- 3.4.Egy adott állapot elérhetőségét jól állapítja meg: 5 pont
- Programminőség: 20 pont
- Értelmesen sorokra tördelt programszöveg: 2 pont
- Bevezető leírás a struktúrák elejének, végének jelzésével: 4 pont
- Programmegységek tagolása (elválasztó sorok): 2 pont
- Eljárások alkalmazása: 4 pont
- Megjegyzések, magyarázatok: 4 pont
- Beszédes azonosítók: 2 pont
- Állandók (szinonimák) alkalmazása: 2 pont

Általános Iskolások számítástechnikai versenye, 1991-1992

1991. Első forduló

Összesen: 55 pont

1. feladat (11 pont)

- Ha a program $C=i^*A+j^*B$ alakban keresi a megoldást, minden ij -re vizsgálva, akkor: 6 pont
- Ha $C-i^*A B$ -vel való oszthatóságát vizsgálja, akkor: 8 pont
- Összes megoldás megadásáért: 3 pont

2. feladat (6 pont)

- A nem szökőéveket 365 nappal számolja: 2 pont
- A 4-gyel osztható éveket szökőévnek veszi: 1 pont
- A 100-zal osztható, de 400-zal nem osztható évek nem szökőévek: 1 pont
- 7-tel való osztás maradéka alapján jól határozza meg a napot: 2 pont

3. feladat (12 pont)

- 10-nél kisebb számok kiírása: 3 pont
- 10-nél nagyobb számok egyes helyértéke: 3 pont
- 10-nél nagyobb számok tízes helyértéke: 3 pont
- 0-ra végződő számok kiírása: 3 pont

4. feladat (8 pont)

- A) Megoldás jó: 2 pont
- B) Megoldás jó: 3 pont
- C) Megoldás jó (hibás válasz esetén levonás!): -4 pont
- B) Magyarázata: 3 pont

5. feladat (18 pont)

- Eredmények beolvasása: 2 pont
- Jó csapatnál számol: 2 pont
- Győzelmek számolása: 2 pont
- Döntetlenek számolása: 2 pont
- Győzelmek számolása: 2 pont
- Lőtt gólok számolása: 2 pont
- Kapott gólok számolása: 2 pont
- Összpontszám számolása: 2 pont
- Eredménytáblázat kiírása: 2 pont

1991. Második forduló

Összesen: 70 pont

1. feladat (7 pont)

A megoldás például (más megoldás is elfogadható):

```

10 FOR I=1 TO 9
20 FOR J=0 TO 9
30 FOR K=0 TO 9
40 IF I*I+J*J+K*K=I*100+J*10+K THEN PRINT I*100+J*10+K
50 NEXT K
60 NEXT J
70 NEXT I

```

- Jó ciklusszervezés: 3 pont
- Jó feltételvizsgálat: 3 pont
- Jó kiírás: 1 pont

2. feladat (11 pont)

1. Egyenlő oldalú háromszöget, oldalhossza 100 egység: 3 pont
 2. Négyzetet, oldalhossza 100 egység: 3 pont
 3. Szabályos 360-szöget, oldalhossza 1 egység, ami körnek látszik: 3 pont
- ami körnek látszik: 2 pont

3. feladat (13 pont)

- Mozog az autó: 1 pont
- A bal oldali lánctalp mozog, ha a fény jobbról jön: 4 pont
- A jobb oldali lánctalp mozog, ha a fény balról jön: 4 pont
- Mindkettő mozog, ha a fény szemből jön: 4 pont

4. feladat (12 pont)

- 130 PRINT U-V 3 pont
- 140 V=U 3 pont
- 150 H=H+D(I)-D(I-1) 3 pont
- az új 150-es sort át kell rakni a ciklus elejére, pl. 115-ösként: 3 pont

5. feladat (12 pont)

- A 120-as sorból az osztást ki lehet emelni a ciklus utánra: 3 pont
- A 160-as sorból az osztást ki lehet emelni a ciklus utánra: 3 pont
- A 160-as sorban A(I)-S négyzetreemelés helyett önmagával szorozható: 3 pont
- A 160-as sorban szorzás esetén a különbséget nem kell kétszer kiszámítani: 3 pont

6. feladat (15 pont)

- 7 szám véletlenszerű kisorsolása: 3 por
- Jó a nyereségszámítás feltétele: 9 por
- Jó a nyereség összege: 3 por

1991. Harmadik forduló

Összesen: 50 pont

- Az útkereszteződés kirajzolása: 5 por
- Paraméterek beolvasása (autók belépése): 5 por
- Az autók vízszintesen mozognak: 6 por
- Az autók függőlegesen mozognak: 6 por
- Véletlenszerűen belépnek vízszintesen: 3 por
- Véletlenszerűen belépnek függőlegesen: 3 por
- A vízszintes út végén eltűnnek: 1 por
- A függőleges út végén eltűnnek: 1 por
- Négyféle lámpát kezel (piros, piros + sárga, zöld, sárga): 4 por
- A lámpaváltások időzítése (a sárga rövid): 4 por
- Jól kezeli az ellentétes irányú lámpát: 2 por
- Az autók piros és sárga lámpánál megállnak: 5 por
- Álló autó mögött a következők megállnak: 5 por

1991. Országos döntő

1991. április 19.

Összesen: 140 pont

1. feladat (45 pont)

- Választás a lehetőségekből: 2 por
- Beolvasás, kiírás: 3 por
- 10 -> 2: 10 por
- 2 -> 10: 10 por
- 16 -> 2: 10 por
- 2 -> 16: 10 por

2. feladat (25 pont)

- Beolvasás, kiírás: 3 por
- Felismer két kezdőbetűt: 10 por
- Felismer tetszőleges számút: 5 por
- Felismeri a kettős mássalhangzókat (cs, gy, ly, ny, sz, ty, zs): 7 por

3. feladat (25 pont)

- J billentyű kezelése: 5 por
- B billentyű kezelése: 5 por
- L billentyű kezelése: 5 por
- F billentyű kezelése: 5 por
- Kezdetben üres a kép, közéről indut: 5 por

4. feladat (45 pont)

- Holdtötek számolása: 10 pont
- Jó holdtölte megtalálása: 5 pont
- Ezt követő vasárnap megtalálása: 20 pont
- Szökőévek kezelése: 5 pont
- Beolvasás, kiírás: 5 pont

1992. Első forduló

Összesen: 100 pont

1. feladat (15 pont)

- Megfelelő számjegyek összeadása: 6 pont
- Ha 10-nél nagyobb, akkor átvitelt számol: 3 pont
- Az átvitelt hozzáadja a következő számjegyhez: 3 pont
- $C(N+1)$ -et is kiszámolja (0 vagy 1 lesz): 3 pont

2. feladat (28 pont)

- Kurzormozgató billentyűk kezelése: 2-2-2-2 pont
- Kurzorkép-választó billentyűk kezelése: 2-2-2-2 pont
- Kurzorképek jó kirajzolása: 3-3-3-3 pont

3. feladat (22 pont)

- Bástya ütésének vizsgálata a bástya sorában: 3 pont
- Bástya ütésének vizsgálata a bástya oszlopában: 3 pont
- Futó ütésének vizsgálata az egyik átlóban: 4 pont
- Futó ütésének vizsgálata a másik átlóban: 4 pont
- Huszár ütésének vizsgálata a 8 lehetséges helyen: 8 pont

4. feladat (21 pont)

- A. A betű: 3 pont
 20 egységnyi szárákkal: 2 pont
 közepén 10 egységnyi összekötő vonal: 2 pont
- B. E betű: 3 pont
 vízszintes vonalai 10 egységnyiek: 2 pont
 függőleges vonala 20 egységnyi: 2 pont
- C. H betű: 3 pont
 vízszintes vonala 10 egységnyi: 2 pont
 függőleges vonalai 20 egységnyiek: 2 pont

5. feladat (14 pont)

- A perct 59 helyett 69 másodperc után növeli (1030 -ban $A(I) \leq 5$ -öt kellene vizsgálni): 4 pont
 Ugyanez a hiba az óra-perc pár hasonló állásánál: 1 pont
- Percváltáskor a másodperc tízes helyértékét nem nullázza (az 1040 -es sorból hiányzik az $A(I)=0$): 4 pont
 Ugyanez a hiba az óra-perc pár hasonló állásánál: 1 pont
- Az óra csak 23-ig növekedhetne, utána nullázni kellene: 4 pont

1992. Második forduló

Összesen: 100 pont

1. feladat (24 pont)

- Megfelelő számjegyek összeszorzása: 8 pont
- Az eredménybe jól számítja be a részletszorzatokat: 10 pont
- Ha 10-nél nagyobb, akkor átvitelt számot: 3 pont
- Az átvitelt hozzáadja a következő számjegyhez: 3 pont

2. feladat (18 pont)

- A. Mitsubishi autóembléma:
- 3 db: 1 pont
 - x élhosszúságú: 3 pont
 - rombusz: 3 pont
 - 120-120 fokkal elforgatva: 2 pont
- B. Audi autóembléma:
- 4 db: 1 pont
 - 10 egységnyire egymásbafonódó: 3 pont
 - 360 egység kerületű: 2 pont
 - kör: 3 pont

3. feladat (16 pont)

- Magas magánhangzók felismerése: 8 pont
- Mély magánhangzók felismerése: 6 pont
- Mássalhangzóhoz nem rendel hangrendet: 2 pont

4. feladat (18 pont)

A ciklus belsejében négy szorzás-osztás, továbbá egy hatványozás van, ezek számát kell csökkenteni.

- Hatványozás megszüntetése: 6 pont
- Szorzások-osztások megszüntetése egyenként: 5-5 pont

5. feladat (10 pont)

- Az autót egy zárt részbe helyeztük, ahonnan nem lehet kijutni: 5 pont
- Az autót egy olyan részbe helyeztük, ahol a "bal kéz a falon" szabállyal körbe lehet járni: 5 pont

6. feladat (14 pont)

- A1. "Általános iskola": 2 pont
- A2. "Fallal bekerített terület": 2 pont
- A3. "13"+CHR\$(0)+CHR\$(7)+"-szor ismételd meg.": 4 pont

- B. Szöveget tömörít: 4 pont
 azaz a 2-nél többször ismétlődő karaktereket a karakter + 2 pont
 CHR\$(0) + CHR\$(darabszám) hármassal helyettesíti:

1992. Harmadik forduló

Összesen: 80 pont

1. feladat (20 pont)

- A háromszög adatainak beolvasása: 3 pont
- Ellenőrzés: "ez tényleg háromszög": 3 pont
- Háromszög megrajzolása: 3 pont
- Jó sorokban húz satírozó vonalat: 3 pont
- A satírozást a sorban jó helyen kezdi: 4 pont
- A satírozást a sorban jó helyen fejezi be: 4 pont

2. feladat (40 pont)

- Beolvasás ellenőrzéssel: 7 pont
- Jól tördel sorokra: 10 pont
- Bal margóhoz igazít: 3 pont
- Jobb margóhoz igazít: 4 pont
- Középre igazít: 4 pont
- Mindkét margóhoz igazít: 12 pont

3. feladat (20 pont)

- Beolvasás: 3 pont
- A hónapokban a napok számát jól kezeli: 4 pont
- A februárt szökőévben is jól kezeli: 3 pont
- A hónap első napjának a jó nevet adja: 5 pont
- Az év első napjának a jó nevet adja: 5 pont

Függelék

Tudnivalók az 1993. évi Nemes Tihamér Országos Középiskolai Számítástechnikai Tanulmányi Versenyről

Tisztelt Kollégák!

Örömmel vettük tanulói jelentkezését a Nemes Tihamér versenyre. Ezúton tájékoztatjuk Önöket az 1993. évi verseny további menetéről.

Ebben az évben is a korábbi gyakorlat szerint szervezzük a versenyt.

a) Kétféle feladatsor készül, egy könnyebb az elsős-másodikos, és egy nehezebb a harmadikos-negyedikes-ötödikes versenyzők számára.

b) Elvárjuk, hogy a versenyzők ismerjék a Pascal nyelvet, és ne legyenek meglepve, ha néhány más programozási nyelven (BASIC, ELAN, LOGO, PROLOG, assembly, pszeudonyelv stb.) is adunk feladatokat; kevésbé ismert nyelvek esetén a program-példákhoz magyarázatot fűzünk.

c) A feladatlapokat és az értékelési útmutatókat nem egyszerre postázzuk.

d) Az iskolai foduló – a többi tanulmányi versenyhez hasonlóan – 14 órakor kezdődik.

e) A döntőben egy vagy két feladatot kell majd megoldani, esetleg már részben megírt programokat kell folytatni, továbbfejleszteni.

1. rész: Szervezési tudnivalók

(1) A hivatalos versenyfelhívás a Közművelődés c. folyóirat 1992. okt. 30-i és a Művelődési Közlöny késő őszi számában jelent meg. Eszerint a verseny első fordulóját **1993. január 12-én, kedden 14.00 órától** kell megtartani az iskolákban. A verseny **időtartamát** a versenybizottság **3 órában** határozta meg. Az iskolai versenyen számítógépet vagy kalkulátort **nem szabad**, mindenféle írásos anyagot (tankönyveket, kézikönyveket, jegyzeteket) szabad használni.

A versenyfelhívás szerint a Nemes Tihamér verseny *első tíz helyezettje az érettségi vizsgán matematika tárgyból jeles osztályzatot kap, továbbá matematikából és számítástechnikából mentesül a felvételi vizsga alól és megkapja a maximális 30 pontot, ha olyan felsőfokú oktatási intézménybe jelentkezett felvételeire, ahol a matematika és/vagy a számítástechnika az felvételi tárgy.* Értelemszerűen ez csak a harmadikosokra, negyedikesekre (és néhol ötödikesekre) vonatkozik.

(2) Az idén 220 iskolából kb. 1720 elsős-másodikos és kb. 1880 harmadikos-negyedikes-ötödikes tanuló jelentkezett eddig a versenyre. E tájékoztatót tartalmazó borítékra – az NTKSztV szót követően – ráírtuk, hogy adataink szerint iskolájukból hány elsős-másodikos (EM), ill. harmadikos-negyedikes-ötödikes (HN) tanuló indulását jelezték. Néhány esetben nem tudjuk, hányadikosok a jelentkezők, ezt IS-sel (ismeretlen) adtuk meg a borítékon. Kérjük, ellenőrizték a számokat, és szükség esetén postafordultával küldjék meg a helyes adatokat a Neumann János Számítógéptudományi Társaság titkárságára (1054 Budapest, V. Báthori u. 16., tel.: (1) 132-9349, (1) 132-9390, fax: (1) 131-8140; **Postacím: NJSZT Nemes Tihamér Verseny, 1368 Budapest 5, Pf. 240).**

(3) Az első forduló feladatsorait legkésőbb 1993. jan. 6-án, szerdán postázzuk. Ha a feladatlapok 1993. jan. 11-éig nem érkeznének meg az iskolába, kérjük, jelentkezzenek értük az NJSZT titkárságán. **A feladatokat tartalmazó borítékot csak a verseny kezdetén, a tanulók előtt szabad felbontani.**

(4) Az értékelési útmutatókat 1993. jan. 12-én, azaz a verseny napján postázzuk. Ha egy hét alatt sem érkeznének meg az iskolába, kérjük, jelentkezzenek értük az NJSZT titkárságán.

(5) A kijavított és legalább az összpontszám 40%-ára értékelt feladatlapokat, a dolgozatkiértékelő lapokat és az iskolai összesítő lapot **1993. jan. 29-éig** kell majd ajánlott levélben postára adniuk. A borítékra a (2)-es pontban megadott címet írják.

(6) Az első fordulóban elért pontszámokat, valamint a második fordulóba bejutottak nevét tartalmazó levelet legkésőbb 1993. márc. 5-éig postára adjuk. A levélben az 1993. márc. 13-án, szombaton megtartandó második forduló, valamint az ünnepélyes eredményhirdetés pontos helyét és idejét is közölni fogjuk.

Tájékoztatásul közöljük, hogy a verseny várható helyszíne a Neumann János Számítástechnikai Középiskola és az ELTE Természettudományi Kara lesz. Az ünnepélyes eredményhirdetést terveink szerint 1993. ápr. 9-én, pénteken 14 órakor tartjuk az NJSZT nagytermében.

2. rész: Tartalmi követelmények

A verseny elsődleges célja, hogy a középiskolák tanulóinak lehetőséget adjon számítástechnikai, főként programozási ismereteik és képességeik összehasonlítására. (Az elsős-másodikosok versenyében természetesen fiatalabbak is indulhatnak.) Egyúttal szeretnénk segítséget adni a számítástechnika iránt érdeklődő tanulóknak és tanáraiknak az iskolai foglalkozások tematikájának összeállításához. A versenybizottság nyílt, szívesen látjuk mindazok észrevételeit, kritikáját, részvételét a versenyek céljának meghatározásában, előkészítésében és megszervezésében, akik ezt társadalmi munkában vállalják, és akik maguk semmilyen módon nem érdekeltek a versenyben.

A verseny jelenleg kétfordulós. Az iskolákban tartandó, 3 órás első fordulóban a tanulók **analizáló** képességét tesszük próbára számítógép használata nélkül: 6-10 kisebb feladatot (algoritmus- vagy programrészletet, működési vázlatot) adunk, és olyan kérdésekre várunk választ, mint pl. (1) mit csinál? (2) milyen hibák vannak benne? (3) milyen feltételek mellett működik? (4) mi hiányzik belőle? (5) mire használjuk a változókat? (6) megoldja-e a kitűzött feladatot? stb.

Az első fordulóból a legeredményesebb 70-100 tanuló jut tovább. A második fordulóban egy vagy két nagyobb, esetleg már részben megoldott, **konstruáló, szintetizáló** jellegű feladatot kell megoldani, folytatni, továbbfejleszteni; a rendelkezésre álló idő 5-6 óra. A versenyzőknek e nagyobb lélegzetű feladat(ok) megoldását meg kell tervezniük, a programo(ka)t meg kell írniuk, számítógépet használva fel kell éleszteniük, és végül dokumentálniuk kell. A megoldandó feladat valamilyen módon a számítástechnikai szakmához és a mai, hazai számítástechnikai helyzethez kötődik. Olyan gépek és nyelvek jöhetnek szóba, amelyekből viszonylag sok van az iskolákban, ill. hazánkban, és amelyeket a verseny helyszínén megfelelő darabszámban a versenyzők rendelkezésére tudunk bocsátani (tehát 1993-ban: IBM-PC/XT és AT, valamint a Pascal és esetleg a BASIC nyelv; más nyelvek a versenybizottság előzetes engedélyével használhatók). Azt szeretnénk, hogy a lehetőségekhez képest egyenlő eséllyel induljanak a versenyzők.

A megoldás(ok) kiértékeléséhez be kell adni az algoritmikus megoldás tervét, a program(ok) szövegét kinyomtatva és gépi adathordozón, a változók szerepének magyarázatát és a program(ok) használati utasítását.

Arra törekszünk, hogy a versenyfeladatok a problémamegoldó, algoritmizáló, modellalkotó, modularizáló készséget mérjék fel. A Pascal vagy a BASIC nyelv használatában való jártasságot elvárjuk a versenyzőktől, de fontosnak tartjuk más programnyelvek (LOGO, ELAN, PROLOG, assembly stb.) szemléletmódjának ismeretét is. A hangsúly nem az egyes nyelvek részleteinek, hanem a módszeres programozás fogalmainak, elveinek és gyakorlatának, a helyes programozási módszereknek és stílusnak géptől és nyelvtől független ismeretén van. A számítógépek és a perifériák felépítésének és működésének rendszerszintű ismeretét szintén elvárjuk.

Reméljük, hogy a versenyen indulók nagy része a számítástechnikai és informatikai módszerek kiváló alkalmazójává válik majd, de a versenyen természetesen nem várjuk el, hogy a versenyzők egyes szakterületek specialistái legyenek.

A szükséges ismeretek vázlatosan:

- A számítógép főbb egységei, perifériás berendezései. Tárolási és adat-rögzítési alapfogalmak.
- BASIC parancsok és utasítások. A Pascal nyelv elemei. Szintaxisábrák, BNF-jelölés. Strukturált vezérlési szerkezetek: felsorolás (szekvencia), választás (elágazás), ismétlés (ciklus). Adattípusok: egész, valós, logikai, szöveg. Összetett adatok: tömb, rekord, lista, verem, sor, fa, állomány stb. Láncolt ábrázolás.
- Fölülről lefelé haladó programozás lépésenkénti finomítással. Programok élesztése, tesztelése, hatékonysági megfontolások. A helyes programozási stílus, a jó program jellemzői (barátságos, biztonságos, módosítható, olvasható, jól dokumentált stb.).
- Számábrázolás bináris, oktális, decimális, hexadecimális számrendszerben. Átalakítások, alapműveletek. Fixpontos és lebegőpontos ábrázolás. Pontosság, túlcsordulás.
- Boole-algebrai és matematikai logikai alapismeretek. Gyökkeresési módszerek. Megoldóképletek. Közeliítő módszerek. Görbe alatti terület kiszámítása. A valószínűségszámítás alapelemei: Gyakoriság, relatív gyakoriság, középérték, súlyozott középérték, hisztogram stb. Véletlenszámok és alkalmazásuk (kockadobás, lottószámok húzása stb.).
- Rendezési eljárások. Adatok beszúrása, törlése, keresése.
- A képernyő grafikus kezelésének alapjai.
- Táblázat- és adatbáziskezelési, szövegszerkesztési ismeretek.

A felkészülés megkönnyítésére a ma már gazdag könyvkínálatból a versenybizott-ság a következőket ajánlja a versenyzők figyelmébe:

1. Alapvető számítástechnikai és programozási ismeretek

- 1.1. Szilávi P., Zsakó L.: Módszeres programozás, MK, Bpest, 1986.
- 1.2. Lócs Gy.: A BASIC és a Kíváncsi (középiskolai szakköri füzet), TK, Bpest, 1985.
- 1.3. N. Wirth: Algoritmusok + adatstruktúrák = programok, MK, Bpest, 1982.
- 1.4. Csépai J.: A számítástechnika alapjai, MK, Bpest, 1985.
- 1.5. Számítástechnika középfokon, OMIKK, Bpest, 1987.
- 1.6. J. Hvorecky, J. Kelemen: Ötlettlő az algoritmusig (középiskolai szakköri füzet, TK, Bpest, 1987.
- 1.7. dr. Urbán J.: Matematikai logika (középiskolai tankönyv), TK, Bpest, 1987.
- 1.8. Simonovits M.: Számítástechnika (középiskolai tankönyv), TK, Bpest, 1985.
- 1.9. Varga T.: Matematikai logika I-II., TK, Bpest, 1960. és 1966.
- 1.10. C.H.A. Koster: Programozás felülnézetben, MK, Bpest, 1988.
- 1.11. Gács P., Lovász L.: Algoritmusok, TK, Bpest, 1987.
- 1.12. D.E. Knuth: A számítógép-programozás művészete. 1.-3. köt., MK, Bpest, 1987-88.

2. Nyelvek

- 2.1. K. Jensen, N. Wirth: A Pascal programozási nyelv, MK, Bpest, 1982 és 1988.
- 2.2. Gordon E., Körtvélyesi G., Sós I., Székely Z.: Pascal programozási nyelv, SZÁMALK, Bpest, 1987.
- 2.3. Bakos T.: Pascal PC-seknek, MK, Bpest, 1986.
- 2.4. Áts L.: Oxford Pascal C64-en, Novotrade Rt, Bpest, 1987.
- 2.5. Turcsányiné Szabó M., D. Senftleben: A LOGO programozási nyelv, MK, Bpest, 1986.
- 2.6. BASIC: lásd 1.2.
- 2.7. ELAN: lásd 1.10.

3. Alkalmazások

- 3.1. Dr. Kis O., Kovács M.: Numerikus módszerek (Bolyai sorozat), MK, Bpest, 1973.
- 3.2. R. Newman, Sproull: Interaktív számítógépes grafika (első fele), MK, Bpest, 1985.
- 3.3. A. Kaufmann: Az optimális programozás (Modellek és módszerek), MK, Bpest, 1968.
- 3.4. Dr. Filep L.: Játékelmélet (középiskolai szakköri füzet), TK, Bpest, 1985.
- 3.5. J. Nievergelt, J.C. Ferrar, E.M. Reingold: Matematikai problémák megoldásának számítógépes módszerei, MK, Bpest, 1977.

4. Példatárak

- 4.1. Lócs Gy.: A BASIC és a Kíváncsi. Feladatgyűjtemény (középiskolai szakköri füzet), TK, Bpest, 1986.
- 4.2. Appel Gy., Kóhegyi J., Zsakó L.: Számítástechnikai feladatok, Interpress, Bpest, 1986.
- 4.3. Számítástechnikai feladatok 2000-ig I-II. OMIKK, Bpest, 1988.
- 4.4. Hanák D. Péter: Programozás ELANNal, MK, Bpest, 1988.
- 4.5. Programozási versenyfeladatok tára. Nemes Tihamér OKSZTV 1985–1992 és Általános iskolások országos számítástechnikai versenye 1991–1992. NJSZT, Budapest, 1992.

a Nemes Tihamér OKSZTV országos versenybizottsága

Megjelent a

Programozási versenyfeladatok tára

a Nemes Tihamér OKSZTV 1985–1992, valamint az Általános iskolások országos számítástechnikai versenye 1991–1992 közötti feladatsoraival kb. 140 oldalon, 120,- Ft-os áron.

A feladattár megvásárolható vagy átvehető a Neumann János Számítógéptudományi Társaság titkárságán (Budapest, V. Báthori u. 16.)

a helyszínen készpénzzel fizetve a pénztári órák alatt, azaz keddtől csütörtökig 9–13 óra között; vagy

a postai befizetést igazoló csekk bemutatásával a titkársági órák alatt, azaz hétfőtől csütörtökig 8–18 és pénteken 8–14 óra között (csekk mellékelve); ill.

írásban postai szállításra, a postai befizetést igazoló csekk egyidejű megküldésével megrendelhető az NJSZT-től a 1368 Budapest 5. Pf. 240. címen (csekk mellékelve). A postaköltséget utánvétellel a megrendelő fizeti.

MTESZ - egyesületi használatra !

**Kiadja: Neumann János Számítógéptudományi
Társaság**

ISBN 963 8431 776

Készült: 500 példányban

513/93 MTESZ Házinyomda, Budapest.

Felelős vezető: Boncza Gábor



