

MA/AIφ
H1φ



SZÁMELMÉLET
nevű számítógépes
oktatási program

ALKALMAZÁSI SEGÉDLET

Dr Perge Inre

=====

Tartalomjegyzék

1. A program készítő adatai
2. A program adatai
3. Elméleti alapok
4. A program használata
5. Minta alkalmazás

=====

1. A program készítő adatai

Név : Dr Perge Imre főiskolai tanár
Intézmény neve: Ho Si Minh Tanárképző Főiskola
Számítóközpontja
Intézmény címe: Eger. Pf.: 43. 3301
Telefonszám : 36 12-399

2. A program adatai

- A program azonosítója (neve): SZAMELMELET
amelyből az első 8 betűt kell megadni a program
betöltéséhez.
- Programnyelv: BASIC
- Memóriaigény: 7005 byte
- Futásidő: A prímszámokat előállító program átlagosan
becsült ideje: $0,07 N$ sec, ahol N a meghatározandó
prímszámok felső korlátja. A többi programrész futás-
ideje 2-10 sec között váltakozik.
- A program készítésének ideje: 1983 március 15.
- Perifériaigény: nincs

3. Elméleti alapok

A számelmélet a matematika egyik legrégebbi ága. A szám-
elméletben több ma is gyakran emlegetett algoritmus még
az időszámításunk előtti korból származik: például Euk-
lidesz algoritmusai, Eratoszthenesz szitája stb.

A számelmélet problémáinak körülhatárolása elég nehéz,
mivel ez a matematika más ágaival szorosan összefonó-
dott. Mi a számelmülethez döntően azokat a problémákat
soroljuk, amelyek az egész számok szerkezeti felépíté-
sével foglalkoznak a középiskolában. Ehhez kívánunk
segítséget adni ezzel a programmal.

=====

Megjegyezzük azonban, hogy egyre többen használják ki a számítógép műveleti gyorsaságát oly módon, hogy meg nem oldott számelméleti problémákat számítógépre irt algoritmusok segítségével közelítenek meg. (Pl. bizonyos tulajdonsággal rendelkező számok, számpárok keresése stb.)

A program lényegében egy programcsomag, amely együtt tartalmazza az általános és középiskolában tanul legfontosabb elemi számelméleti témakörök (feladatok) programjait. Célja, hogy segítse a számolásigényes számelméleti feladatok megoldását és a természetes számok strukturájának a vizsgálatát.

A program a matematika tantárgy tanításához készült, de jól felhasználható a számítástechnikai szakkörök keretében is, az alábbi számelméleti feladatok megoldására:

- számok átalakítása egyik számrendszerből egy másik számrendszerbe
- prímszámok előállítása és száma egy adott számig
- egy szám valódi osztói (törzsoztói)
- egy szám törzstényező felbontása
- legnagyobb közös osztó meghatározása (lnko)
- legkisebb közös többszörös meghatározása (lkkt)

A program felhasználható a fenti részfeladatok oktatásánál és gyakorlatánál, az egyes fogalmak sokoldalú megerősítéséhez, újabb problémák vizsgálatához (pl. lnko és lkkt kapcsolata, 2-es, 4-es, 8-as, 16-os számrendszer kapcsolata, törzstényező felbontás és lnko stb). Jól felhasználható továbbá ilyen jellegű feladatok megoldásának az ellenőrzésére is, és lehetővé teszi, hogy szinte minden tanulónak más-más feladatot adjunk, tekintettel

=====

arra, hogy a megoldásra fordított idő a prímszámok meghatározásának kivételével szinte csak a néhány bemenő adat begépelésével azonos.

A fenti számelméleti feladatok és fogalmak valamennyi középiskola matematikai tantervi anyagában előfordulnak és ezekhez szervesen kapcsolódnak. A Számítástechnika szakkörökön a számrendszerek átalakítása jól hasznosítható, az egyes programrészek tanulmányozása és önálló programként való megírása pedig feladatként adható.

Az első programszegmens jól felhasználható a 2-től 16-ig terjedő számrendszerbeli számok előállítására, a számrendszerek közötti összefüggések és számrendszerbeli műveletek vizsgálatára.

A második programszegmens azon kívül, hogy adott számig előállítja a prímszámokat, lehetőséget ad a prímszámok számának és egyéb tulajdonságainak a vizsgálatára is.

A harmadik programrész egy szám valódi osztóit (törzsoosztóit) állítja elő és felhasználható ezek számának a vizsgálatára is.

A negyedik programszegmens a számok törzstényezőinek és azok kitevőinek a meghatározására szolgál és felhasználható a \lnko és $lkkt$ kiszámítására is.

Az ötödik és hatodik programrész két vagy több szám \lnko .-jának illetve $lkkt$.-ének a kiszámítására használható. Korlát csak a $lkkt$ -nél van, ugyanis az nem lehet nagyobb a gépben tárolt maximális egésznel.

=====

4. A program használata

A program kazettás mágnesszalagra rögzített és a kazettán feltüntetett számláló állásnál kezdődik. A programot e számláló állástól a szalagról a

LOAD CAS: (HT gépen CLOAD)

parancs begépelésével tölthetjük be, és a
RUN

parancsra indíthatjuk el, amelynek hatására a képernyőn megjelenik a program működésével kapcsolatos tájékoztató.

Ezután a program 1-7-ig terjedő szám begépelésére vár, amelyet a RETURN billentyűvel zárunk. Nem megfelelő szám leütése esetén tovább várakozik a helyes szám begépelésére.

E szám hatására a program a megfelelő programrészre kapcsol. Egy-egy programrész futása végén a program egy billentyű leütésére várakozik. Ha ilyenkor a "V" betűt ütjük le (vége), akkor a program visszatér a tájékoztatóhoz és újabb részfeladat megoldására várakozik. Minden más billentyű leütése esetén az előző feladattípus megoldására várakozik.

Valamennyi részfeladat futásánál a képernyő törlődik, és kiírásra kerül a feladat típusa, majd a hozzátartozó bemenő adatok megnevezése, amely után a program a kért adat begépelésére vár. Valamennyi adatot a RETURN billentyűvel kell zárni. Hibás adat esetén vagy az adat ismétlését kéri, vagy a részprogram elejére tér vissza a program és a "jó" adatok begépelésére vár.

A feladatok megoldása után a képernyőn a BEFEJEZTEM szó jelenik meg. HT gépen a RETURN helyett NEW LINE értendő.

=====

Az 1. részprogramnak 3 bemenő adata van:

- az átalakítandó szám, amelynek jegyei
0,1,2,...,9,A,B,C,... lehetnek. A begépelt jegyek száma a számrendszer alapszámától függ:
pl. 2-es számrendszerben 15,
8-as számrendszerben 5 lehet a jegyek száma.
Az átalakítandó szám felső korlátja 32767. Legyen Pl. az átalakítandó szám
16-os számrendszerbeli 7FFF és írjuk át 8-as számrendszerbe.
- az eredeti alapszám, példánkban 16
- az új alapszám, példánkban 8
Mindkettő 2-20 közötti egész tizes számrendszerbeli szám lehet.

Amennyiben a számrendszerhez nem tartozó számjegyet adunk meg a számban, rövid időre kiírásra kerül a "HIBAS JEGY" szöveg és a program visszatér a programrész elejére. A program outputja: Először kiíródik, hogy a szám

A TIZES SZÁMRENDSZERBEN: 32767

majd egyenlőség formájában az eredeti és a kívánt számrendszerbeli szám, a számok mögött zárójelben a megfelelő alapszámok.

Példánkban:

7FFF(16)= 7 7 7 7 7(8)

- A 2. részprogramnak egy bemenő adata van, az a szám, ameddig a primszámokat meg akarjuk határozni. A kimenő információk csak akkor kerülnek kiírásra, ha már valamennyi primszámot előállított a program a kívánt számig. A kiírás 6 oszlopos és 10 soros blokkokban kerül kiírásra. Ilyenkor ha szükséges, tetszőleges billentyű leütésével aktivizálható a további blokkok kiírása, mindaddig amíg meg nem jelenik a kiírás vége, hogy hány db primszám van.

=====

A 3. részprogramnak egy bemenő adata van, az a szám, amelynek a valódi osztóit meg akarjuk határozni. Az output két oszlopban (az első és a 20-as pozíción) kerül kiírásra. Pl. 24-es bemenő adat esetén az output

2 TORZSOSZTÓ	12
3 TORZSOSZTO	8
4	6

A VALODI OSZTOK SZAMA: 6

A TORZSOSZTOK SZAMA: 2

A kiírás 10 soronként itt is leáll és tetszőleges billentyű leütésével tovább íratható ha szükséges.

A 4. típusu feladatnál ugyancsak 1 bemenő adat van, az a szám amelynek a törzstényezőit kívánjuk előállítani. Az outputban kiírásra kerül a TORZSTENYEZO KITEVOJE fejléc, majd alatta a megfelelő számok. Pl. 24 esetén az output a következő

TORZSTÉNYEZO	KITEVOJE
2	3
3	1

Az 5. programszegmens első bemenő adata a számok db száma, amelyeknek a lnko-ját akarjuk meghatározni, majd ezt követik sorban egymás alá gépelve maguk a számok. A számok db száma nagyobb egynél és kisebb 11-nél, maguk a számok pedig nem lehetnek nagyobbak 32767-nél. Az output nemcsak a végeredményt írja ki, hanem az algoritmusnak megfelelően zárójelben az egymás után következő két megfelelő számot és lnko-ját. Pl. a 3,24,36,42 bemenő információk esetén az output

(24 , 36) = 12

(12 , 42) = 6

=====

A 6. programszegmens első bemenő adata a számok db száma, amely itt is nagyobb 1 és kisebb mint 11 lehet, amelyet sorban egymás alá gépelve annyi szám követ, amennyit éppen megadtunk. A számok nagyságrendjét itt nemcsak a gépi 32767 szám korlátozza, hanem az eredmény is, amely nem lehet nagyobb 32767-nél. Amennyiben ez mégis bekövetkezne, a program kijelzi, hogy TUL NAGY A LKKT. és újabb feladat megoldására várakozik. Az output itt is kiírja a közbülső eredményeket is. Pl. a 3,24,36,42 bemenő adatok esetén az output

$$\text{LKKT} (24,36) = 72$$

$$\text{LKKT} (72,42) = 504$$

A program leállítása 7. opció megadásával eszközölhető a program vezérlő részéből, vagy egyéb a gépi reprezentánsban megadott módon.

5. Minta_alkalmazás

- a. Hasonlítsuk össze a 252 és a 372, valamint a legkisebb közös többszörösük törzstényező előállítását és adjuk meg a lkkt. tárbeli, 2-es számrendszerbeli alakját.

Megoldási terv:

1. A 252 és 372 lkkt.-ének meghatározása (6-os feladat)
2. 252 törzstényező felbontása (4-es feladat.)
3. 372 törzstényező felbontása (4-es feladat.)
4. lkkt törzstényező felbontása (4-es feladat.)
5. lkkt a 2-es számrendszerben (1-es feladat.)

=====

A képernyőn megjelenő kérdésekre adott választ aláhúzással, a leütött de a képernyőn láthatatlan karaktereket (billentyűket) bekeretezve jelöljük. A RETURN billentyűt -el rövidítjük.

RUN

parancsra a képernyőn megjelenik a tájékoztató. Először a 6-os számú feladatot kérjük.

.....
 HANYAS SZAMU FELADATOT KERED:?6

Ennek hatására az alábbi képernyőt állítjuk elő, az aláhúzott válaszokkal:

LE GKISEBB KOZOS TOBBSZOROS KISZAMITASA.

HANY DB SZAM VAN?2

A SZAMOK SORBAN EGYMAS ALA IRVA:

?252

?372

LKKT(252, 372)= 7812

BEFEJEZTEM

,

Irjuk le a 7812-öt először, mert majd szükségünk lesz rá a későbbiekben és csak utána üssük le a villogó cursor megjelenésére a V-billentyűt, mivel további ilyen feladat megoldására nincs szükség. Ennek hatására újra megjelenik a képernyőn a tájékoztató. Most a 4-es feladatot kérjük háromszor.

.....
 HANYAS SZAMU FELADATOT KERED:?4

TORZSTENYEZOS FELBONTAS, A SZAM?252 R

TORZSTENYEZO KITEVOJE

2	2
3	2
7	1

BEFEJEZTEM

N

A $2^2 \cdot 3^2 \cdot 7$ -et is jegyezzük fel, és mivel a továbbiakban is ugyanilyen típusu feladatot kívánunk megoldani, ezért a villogó cursor megjelenésére pl. leütjük az N billentyűt (a V kivételével bármelyik lehet) és a további teendő hasonló az előzőhöz.

TORZSTENYEZOS FELBONTAS, A SZAM?372 R

TORZSTENYEZO KITEVOJE

2	2
3	1
31	1

BEFEJEZTEM

H

$$372 = 2^2 \cdot 3 \cdot 31$$

TORZSTENYEZOS FELBONTAS, A SZAM?7812 R

TORZSTENYEZO KITEVOJE

2	2
3	2
7	1
31	1

BEFEJEZTEM

V

=====

A $7812=2^2 \cdot 3^2 \cdot 7 \cdot 31$ törzstényezős előállítás alapján meggyőződhetünk arról a tanult tulajdonságról, hogy a 7812 törzstényezői a 252 és 372 közös és nem közös törzstényezőit tartalmazza az előforduló legnagyobb hatványkitevővel.

Végül az 1-es feladatot kérjük.

.....
HANYAS SZÁMU FELADATOT KERED: ?1 R

SZÁMOK ATALAKITÁSA ÚJ SZÁMRENDSZERBE.

A SZÁM: ?7812 R

AZ EREDETI ALAPSZÁM: ?10 R

AZ ÚJ ALAPSZÁM: ?2 R

A TIZES SZÁMRENDSZERBEN: 7812

7812(10) = 1 1 1 1 0 1 0 0 0 0 1 0 0(2)

BÉFEJEZTEM

V

b. Állítsuk elő az 500-nál kisebb prímszámokat.

Ennél a példánál a 2-es feladatot kérjük.

.....
HANYAS SZÁMU FELADATOT KERED: ?2 R

PRÍMSZÁMOK ELOÁLLITÁSA: MEDDIG ?500 R

(MOST SZÁMOLOK, VARJ TURELMESEN !)

2 3 5 7 11 13

17 19 23 29 31 37

.....
257 263 269 271 277 281 B

.....
467 479 487 491 499

500-IG A PRÍMSZÁMOK SZÁMA= 95

BÉFEJEZTEM

V

=====

Tekintettel arra, hogy a prímszámok kiírásánál még nem értünk el a végére (nem került kiírásra hogy hány db prímszám van) ezért a villogó cursor megjelenése után folytatódik a kiírás. Mi példánkban a B betűt ütöttük le.

- c. Állítsuk elő a 9800 valódi osztóit, majd állítsuk le a programot.

Ennél a példánál először a 3., majd ennek befejezése után a 7. programrészt kérjük.

.....

HANYAS SZÁMU FELADATOT KERED: ?3

VALODI OSZTOK MEGHATÁROZÁSA. A SZÁM:
?9800

2 TORZSOSZTO	4900
4	2450
5 TORZSOSZTO	1960
7 TORZSOSZTO	1400
8	1225
10	980
14	700
20	490
25	392
28	350 <input type="text" value="A"/>
35	280
40	245
49	200
50	196
56	175
70	140
98	100

A VALODI OSZTOK SZÁMA: 34

A TORZSOSZTOK SZÁMA : 3

BEFEJEZTEM

=====

Ennél a példánál is, mivel a cursor még a sor végén villog, nem fejeződött be az összes osztó kiírása. Tetszőleges billentyű leütésére (ez nálunk A) a kiírás folytatódik.