

## Probleme pentru atestat 2008 – Programare

- 1) Se citesc de la tastatură  $n$  fracții în forma (numărător, numitor). Se cere să se calculeze suma acestor fracții (în forma ireductibilă).  
*Ex.*  $\frac{7}{6} + \frac{1}{3} + \frac{1}{4} + \frac{2}{5} = \frac{43}{20}$ .
- 2) Să se scrie un program care calculează „cifra de control” a unui număr întreg efectuând suma cifrelor sale, apoi suma cifrelor acestei sume etc. până se obține o sumă formată dintr-o singură cifră.  
*Ex.* Cifra de control a numărului 1971 este 9 (1971  $\rightarrow$  18  $\rightarrow$  9).
- 3) Să se scrie un program care determină primele  $n$  cifre din seria fracției subunitare  $k/m=0.c_1c_2\dots c_n$  pentru  $k, m$  ( $k < m$ ) și  $n$  numere naturale citite de la tastatură.  
*Ex.* Pentru  $k=5, m=23$  și  $n=5$  se va afișa 0.21739.
- 4) Să se afișeze toate numerele prime de trei cifre care citite invers, sunt tot numere prime.  
*Ex.* Un astfel de număr este 761 (761 este prim și 167 este tot prim).
- 5) Se spune că  $n$  este deosebit dacă există un număr natural  $m$  astfel încât  $n=m+S(m)$ , unde  $S(m)$  este suma cifrelor lui  $m$ . Să se scrie un program care verifică dacă un număr natural  $n$  dat este deosebit.  
*Ex.* 1235 este deosebit ( $1235=1225+10$ ).
- 6) O carte se paginează cu numere de la 1 la  $n$  ( $20 \leq n \leq 10000$ ,  $n$  natural). Fiind dat  $n$ , să se afle câte cifre au fost folosite la paginare.  
*Ex.* Pentru o carte cu 100 de pagini se folosesc 192 de cifre.
- 7) Să se determine dacă două numere naturale date sunt termeni consecutivi ai șirului Fibonacci.  
*Ex.* 55 și 89 sunt termeni consecutivi în șirul Fibonacci.
- 8) Să se scrie un program care determină toate reprezentările posibile ale unui număr natural ca sumă de numere naturale consecutive.  
*Ex.* Pentru  $n=50$  avem  $50=8+9+10+11+12$  și  $50=11+12+13+14$ .
- 9) Se numesc numere „super prime”, acele numere ale căror prefixe sunt toate numere prime. Să se verifice dacă un număr este „super prim” și să se afișeze prefixele sale în acest caz.  
*Ex:* 239 este super prim deoarece 2, 23 și 239 sunt prime.
- 10) Se dă  $n$  număr natural. Să se afișeze cel mai apropiat număr prim față de  $n$ . Programul va conține cel puțin un subprogram.
- 11) Să se afișeze câte din elementele unui vector dat de numere naturale sunt numere Fibonacci.  
*Ex.* În vectorul  $x=(5, 10, 1, 7, 9, 8, 1, 6, 55, 19)$  sunt 5 numere Fibonacci.
- 12) Să se verifice dacă un vector dat este o mulțime (în sensul cunoscut din matematică), iar dacă nu este, să se transforme acest vector în mod corespunzător.  
*Ex.* Vectorul  $x=(1, 2, 9, 4, 2, 6, 5, 1, 20, 3)$  nu este o mulțime. După transformarea acestuia se obține mulțimea  $x=(1, 2, 9, 4, 6, 5, 20, 3)$ .
- 13) Se citesc de la tastatură doi vectori cu elemente numere întregi de dimensiune  $n$ , respectiv  $m$ . Să se afișeze câte elemente din al doilea vector sunt strict mai mici decât toate elementele din primul vector.
- 14) În fișierul ‘DATE.TXT’ se găsesc două numere mari (care pot avea mai mult de 10 cifre), câte unul pe o linie. Să se afișeze suma celor două numere.  
*Ex :* pentru fișierului ‘DATE.TXT’  
45899200768797  
50999867  
se va afișa 45899251768664
- 15) Se consideră un vector cu  $n$  componente. Știind că el conține două subsecvențe de numere ordonate crescător, să se ordoneze întregul vector prin interclasarea celor două subsecvențe.
- 16) Se citește de la tastatură un tablou unidimensional cu  $n$  elemente numere întregi. Să se afișeze elementul care apare de cele mai multe ori în tablou. Dacă există mai multe astfel de elemente, se vor afișa toate.  
*Ex.* Pentru  $n=8$  și elementele (23, 7, 11, 7, 19, 7, 11, 11) se vor afișa elementele 7 și 11, care apar fiecare de câte 3 ori.
- 17) Fie  $v$  un vector de numere întregi. Să se construiască un vector  $w$ , astfel încât  $w[i]=$ numărul de apariții ale lui  $v[i]$  în vectorul  $v$ . Să se afișeze cei doi vectori, fiecare pe câte o linie.  
*Ex.* Pentru  $v=(1, 5, 2, 1, 5, 7, 2, 1, 5)$  se obține  $w=(3, 3, 2, 3, 3, 1, 2, 3, 3)$ .
- 18) Dându-se un vector cu  $n*n$  elemente, se cere să se introducă elementele vectorului într-o matrice pătratică de dimensiune  $n*n$ , prin parcurgerea acesteia în ordinea  $a_{11}, a_{21}, a_{22}, a_{12}, a_{31}, a_{32}, a_{33}, a_{23}, a_{13}, a_{41}, \dots$   
*Ex:* Pentru vectorul ( 1, 4, 5, 2, 7, 8, 9, 6, 3), se va obține matricea

1	2	3
4	5	6
7	8	9.

- 19) Să se construiască o matrice  $A$  cu  $n$  linii și  $n$  coloane ce se completează cu termenii șirului lui Fibonacci. Completarea se va face pe linii. Nu se vor folosi structuri de date auxiliare.

Ex. Pentru  $n=3$  se va afișa matricea :

1	1	2
3	5	8
13	21	34.

- 20) Se dă o matrice de ordinul  $n$ . Se consideră că diagonalele sale împart matricea în 4 zone: nord, sud, vest și est. Se cere să se calculeze suma elementelor impare din zona de nord a matricei.

- 21) Se citește de la tastatură un cuvânt de lungime cel mult 20 de caractere, format numai din litere mari. Să se afișeze toate cuvintele distincte ce se pot forma prin eliminarea câte unui singur caracter din cuvântul dat.

Ex. Pentru cuvântul BINE se vor afișa, nu neapărat în această ordine, cuvintele: INE, BNE, BIE, BIN.

- 22) Se citește de la tastatură un text, precum și două cuvinte  $c1$  și  $c2$ . Să se înlocuiască în text toate aparițiile cuvântului  $c1$  cu cuvântul  $c2$ , iar apoi să se tipărească noul text obținut.

- 23) Se citește de la tastatură un număr natural. Să se afișeze cel mai mare număr care se poate forma cu cifrele distincte ale numărului dat.

Ex. Pentru numărul 29363, mulțimea cifrelor distincte este  $\{2, 3, 6, 9\}$ , iar numărul cerut este 9632.

- 24) Scrieți un program care citește de la tastatură două numere naturale  $n$ ,  $m$  și scrie în fișierul text 'DATE.TXT' toate numerele prime din intervalul deschis  $(n,m)$ . Numerele se scriu în ordine crescătoare, câte 10 numere pe fiecare linie a fișierului, numerele dintr-o linie fiind despărțite între ele prin câte un spațiu.

Ex. Pentru  $n=87$  și  $m=241$ , fișierul DATE.TXT va conține :

```
89 97 101 103 107 109 113 127 131 137
139 149 151 157 163 167 173 179 181 191
193 197 199 211 223 227 229 233 239 .
```

- 25) Pentru un număr natural  $n$  dat, afișați descompunerea lui în factori primi. Se va folosi cel puțin un subprogram în rezolvarea cerinței.

Ex : pentru  $n=48$  se va afișa factorul 2 la puterea 4  
factorul 3 la puterea 1

- 26) Pentru un număr natural  $n$  dat să se construiască recursiv triunghiul de numere ca în exemplul de mai jos fără a folosi nici o instrucțiune repetitivă :

```
1
1 2
1 2 3
1 2 3 4
.....
1 2 3 4 ... n
```

- 27) Un tablou unidimensional  $v$  conține  $n$  numere reale ordonate crescător. Să se afișeze în ce poziție din  $v$  se găsește un număr real  $x$  dat. Dacă nu se găsește în  $v$  atunci să se afișeze un mesaj corespunzător. Se va folosi o metodă eficientă de căutare.

- 28) Fie  $n$  un număr natural. Să se genereze toate succesiunile posibile de  $n$  ( $n < 20$ ) caractere '\*' și '-'.

Ex. Pentru  $n=3$  succesiunile sunt

```
---
--*
-_*
_**
*__
*_*
**_
***.
```

- 29) Să se genereze și să se afișeze toate submulțimile mulțimii  $\{a_1, a_2, \dots, a_n\}$ .

- 30) Să se afișeze toate modurile posibile de a descompune un număr natural  $n$  în sumă de  $k$  numere distincte ( $n$  și  $k$  sunt cunoscute). Soluțiile se vor scrie în fișierul 'REZ.TXT', câte o descompunere pe o linie, sub forma sugerată de exemplu.

Ex. Pentru  $n=9$  și  $k=3$ , soluțiile sunt :

```
1+2+6
2+3+4
1+3+5
```