UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Concursul de admitere (nivel licență) - iulie 2017

Proba scrisă la Informatică

VARIANTA II

În atenția concurenților:

1. Rezolvările se vor scrie în *pseudocod* sau *într-un limbaj de programare* (*Pascal/C/C++).*
2. Primul criteriu în evaluarea rezolvărilor va fi ***corectitudinea*** algoritmului, iar apoi ***performanța*** din punct de vedere al *timpului de executare* și al *spațiului de memorie utilizat*.
3. ***Este obligatorie descrierea și justificarea*** (sub) algoritmilor înaintea rezolvărilor. Se vor scrie, de asemenea, ***comentarii*** pentru a uşura înţelegerea detaliilor tehnice ale soluției date, a semnificaţiei identificatorilor, a structurilor de date folosite etc. Neîndeplinirea acestor cerințe duce la pierderea a 10% din punctajul aferent subiectului.
4. Nu se vor folosi funcții sau biblioteci predefinite (de exemplu: *STL*, funcţii predefinite pe şiruri de caractere).

<u>Subiectul I</u> (35 puncte)

1. Degustare de ciocolată (20 puncte)

O companie de publicitate face reclamă la un nou sortiment de ciocolată și intenționează să distribuie mostre de ciocolată la $\boldsymbol{n}$ ($10 \le \boldsymbol{n} \le 10\,000\,000$) copii care sunt așezați într-un cerc. Angajații companiei își dau seama că distribuirea de mostre tuturor copiilor ar costa foarte mult. În consecință, decid să distribuie mostre fiecărui al $\boldsymbol{k}$-lea ($0 < \boldsymbol{k} < \boldsymbol{n}$) copil din cei $\boldsymbol{n}$, numărând copiii din $\boldsymbol{k}$ în $\boldsymbol{k}$ (atunci când numărătoarea ajunge la ultimul copil, ea continuă cu primul copil și așa mai departe). În numărătoare se vor considera toți copiii, fie că au primit sau nu ciocolată. Numărătoarea *se oprește atunci când o ciocolată ar trebui distribuită unui copil care deja a primit.*

Scrieți un subalgoritm care determină numărul copiilor (***nr***) care nu primesc mostre de ciocolată. Parametrii de intrare sunt numerele naturale $\boldsymbol{n}$ și $\boldsymbol{k}$, iar parametrul de ieșire va fi numărul natural ***nr***.

Exemplu 1: dacă $\boldsymbol{n} = 12$ și $\boldsymbol{k} = 9$, atunci ***nr*** = 8 (primul, al 2-lea, al 4-lea, al 5-lea, al 7-lea, al 8-lea, al 10-lea, al 11-lea copil nu primesc ciocolată).

Exemplu 2: dacă $\boldsymbol{n} = 15$ și $\boldsymbol{k} = 7$, atunci ***nr*** = 0 (toți copiii primesc ciocolată).

2. Reducere (15 puncte)

Se consideră șirurile ***a*** și ***b*** cu $\boldsymbol{n}$ ($1 \le \boldsymbol{n} \le 10\,000$), respectiv $\boldsymbol{m}$ ($1 \le \boldsymbol{m} \le 10\,000$) elemente numere naturale mai mici decât 30 000. O subsecvență a unui șir este formată din elemente ale șirului aflate pe poziţii consecutive în șirul dat. Spunem că șirul ***a*** „*se poate reduce*" la șirul ***b*** dacă există o împărţire a șirului ***a*** în subsecvenţe disjuncte astfel încât:

- prin concatenare, în ordine, a tuturor subsecvențelor se obține șirul ***a***;
- prin înlocuiri, în ordine, a tuturor subsecvenţelor cu suma elementelor lor se obţin, în ordine, elementele șirului ***b***.

Scrieți un subalgoritm care stabilește dacă șirul ***a*** *se poate reduce sau nu* la șirul ***b***. În caz afirmativ, identificați elementul din șirul ***b*** (și poziția ***k*** a acestui element) obținut prin însumarea valorilor din cea mai lungă subsecvență a șirului ***a***. Subalgoritmul are ca parametri de intrare cele două șiruri ***a*** și ***b***, precum și lungimile lor ***n*** și, respectiv, ***m***. Parametrii de ieșire vor fi ***răspuns***, ***k*** și ***nrMax***, unde: ***răspuns*** va avea valoarea ***adevărat*** dacă răspunsul la întrebare este *afirmativ*, respectiv ***fals***, în caz contrar; ***k*** reprezintă indicele elementului din șirul ***b*** care se obține însumând elementele din subsecvența de lungime maximă (***nrMax***). Dacă există mai multe subsecvențe de lungime maximă, se va considera prima dintre ele. Dacă șirul ***a*** nu *se poate reduce* la șirul ***b***, ***k*** și ***nrMax*** vor avea fiecare valoarea -1.

Exemplul 1: dacă $\boldsymbol{n} = 12$, $\boldsymbol{a} = (6, 3, 4, 1, 6, 4, 6, 1, 7, 1, 8, 7)$, $\boldsymbol{m} = 4$ și $\boldsymbol{b} = (13, 7, 18, 16)$, atunci ***răspuns*** = ***adevărat***, deoarece $6 + 3 + 4 = 13$, $1 + 6 = 7$, $4 + 6 + 1 + 7 = 18$, $1 + 8 + 7 = 16$. Astfel, $\boldsymbol{k} = 3$ și ***nrMax*** = 4.

Exemplul 2: dacă $\boldsymbol{n} = 17$, $\boldsymbol{a} = (10, 12, 11, 2, 2, 3, 2, 3, 13, 3, 41, 5, 4, 5, 6, 5, 2)$, $\boldsymbol{m} = 6$ și $\boldsymbol{b} = (33, 4, 15, 41, 25, 2)$, atunci ***răspuns*** = ***fals***, deoarece $10 + 12 + 11 = 33$, $2 + 2 = 4$, dar $3 + 2 + 3 < 15$, iar $3 + 2 + 3 + 13 > 15$, deci valoarea $\boldsymbol{b}_3 = 15$ nu se poate obține însumând elemente consecutive din șirul ***a***.

Notă: În exemplele date șirurile sunt indexate începând cu 1.

Subiectul II (15 puncte)

Se dă următorul subalgoritm unde ***n*** este parametru de intrare, iar ***p*** și ***i*** sunt parametri de ieșire (***n***, ***p***, ***i*** - numere naturale, $1 \le n \le 1\,000\,000$, $0 \le p \le 1\,000\,000$, $0 \le i \le 1\,000\,000$):

```
Subalgoritm f(n, p, i):
   Dacă n ≤ 9 atunci
      Dacă n mod 2 = 0 atunci
           { n mod 2 calculează restul împărțirii  lui n la 2}
         p ← n
         i ← 0
      altfel
         p ← 0
         i ← n
      SfDacă
   altfel
      f(n div 10, p, i)
           { n div 10 calculează câtul împărțirii lui n la 10}
      Dacă n mod 2 = 0 atunci
         p ← p * 10 + n mod 10
      altfel
         i ← i * 10 + n mod 10
      SfDacă
   sfDacă
SfSubalgoritm
```

a. Enunțați problema pe care o rezolvă subalgoritmul dat.

b. Ce valori vor avea ***p*** și ***i*** după apelul `f(205 609, p, i)`?

c. Scrieți o variantă iterativă (ne-recursivă) a subalgoritmului dat respectând antetul subalgoritmului din varianta recursivă.

Subiectul III (40 puncte)

Prelucrări imagine

O imagine alb-negru este reprezentată codificat printr-un tablou bidimensional cu valori 0 (pixel alb) și 1 (pixel negru). Asupra imaginii se pot efectua transformări precum:

- Inversarea (I), adică valorile 0 se transformă în 1 și valorile 1 se transformă în 0;
- Rotirea cu 90 de grade în sensul acelor de ceasornic (R);
- Zoom (Z), adică fiecare pixel va fi expandat în 4 pixeli identici cu cel inițial. $\boxed{1} \rightarrow \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$

O secvență de transformări se definește ca o succesiune de litere I, R și Z (în orice ordine).

Scrieți un program care, fiind dat un tablou bidimensional ***imagine*** având ***m*** linii și ***m*** coloane (***m*** - număr natural, $2 \le m \le 10$) și o secvență ***s*** care conține cel mult cinci transformări, aplică aceste transformări și afișează imaginea obținută în urma transformărilor.

Exemplu*:** dacă ***m = 3, ***imagine*** = $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ și ***s*** = (R, I, R, Z), atunci rezultatul va fi $\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$.

Imaginile intermediare (după aplicarea succesivă a transformărilor R, I, R, Z) sunt:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

În rezolvare folosiți subprograme pentru:

a. citirea datelor de intrare de la tastatură (datele de intrare se consideră corecte în raport cu cerințele);
b. inversarea unei imagini;
c. rotirea cu 90 de grade a unei imagini;
d. aplicarea operației de zoom pe o imagine;
e. afișarea pe ecran a unei imagini.

Notă:

1. Toate subiectele sunt obligatorii.
2. Rezolvările trebuie scrise detaliat pe foile de examen (ciornele nu se iau în considerare).
3. Se acordă 10 puncte din oficiu.
4. Timpul efectiv de lucru este de 3 ore.