UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Concurs de admitere (nivel licență) - sesiunea iulie 2016
Proba scrisă la Informatică
VARIANTA I

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 necesară folosirea ***comentariilor*** pentru a uşura înțelegerea rezolvării date (se va explica semnificația identificatorilor și se vor descrie ***ideile principale pe care se bazează rezolvarea***).
4. Nu se vor folosi funcții sau biblioteci predefinite (de exemplu: *STL*, funcții predefinite pe şiruri de caractere etc.).

<u>Subiectul I</u> (50 puncte)

1. Turnuri (10 puncte)

Se consideră un număr suficient de mare de monede de dimensiuni egale pentru a construi din ele turnuri pe baza următoarelor reguli:

R1. cel mai înalt turn are înălțimea de ***n*** monede ($0 < n \leq 13$), cel mai mic are înălțimea 1 (o monedă);
R2. turnurile se așează unul lângă altul, astfel încât între oricare două turnuri de aceeași înălțime să existe cel puțin un turn mai înalt decât acestea două.

Scrieți un subalgoritm care calculează *numărul maxim de turnuri* (***nrTurnuri***) care se pot construi respectând regulile date, precum și *numărul de monede* (***nrMonede***) necesare construcției. Înălțimea ***n*** a celui mai înalt turn este parametru de intrare al subalgoritmului, iar ***nrTurnuri*** și ***nrMonede*** vor fi parametri de ieșire.
Exemplu: dacă ***n*** = 3, atunci ***nrTurnuri*** = 7 și ***nrMonede*** = 11.

2. Numere magice (20 puncte)

Se consideră două numere naturale ***p*** și ***q*** ($2 \leq p \leq 10$, $2 \leq q \leq 10$). Un număr natural se numește *magic* dacă mulțimea cifrelor utilizate în scrierea lui în sistemul de numerație având baza ***p*** este identică cu mulțimea cifrelor folosite în scrierea lui în sistemul de numerație având baza ***q***. De exemplu, pentru ***p*** = 9 și ***q*** = 7, $(31)_{10}$ este număr *magic* pentru că $(34)_9 = (43)_7$, iar pentru ***p*** = 3 și ***q*** = 9, $(9)_{10}$ este număr *magic* pentru că $(100)_3 = (10)_9$.

Scrieți un subalgoritm care, pentru două baze ***p*** și ***q*** date determină șirul ***x*** al tuturor numerelor *magice* strict mai mari ca 0 și strict mai mici decât un număr natural ***n*** dat ($1 < n \leq 10\,000$). Parametrii de intrare ai subalgoritmului sunt ***p*** și ***q*** (cele două baze) și valoarea ***n***. Parametrii de ieșire vor fi șirul ***x*** și lungimea ***k*** a șirului ***x***.
Exemplu: dacă ***p*** = 9, ***q*** = 7 și ***n*** = 500, șirul ***x*** va avea ***k*** = 11 elemente: (1, 2, 3, 4, 5, 6, 31, 99, 198, 248, 297).

3. Inserare (20 puncte)

Fie ***a*** un şir cu ***n*** ($0 < n \leq 10\,000$) numere naturale strict pozitive strict mai mici decât 30 000.

Scrieți un subalgoritm care inserează după fiecare element al șirului un nou număr care reprezintă cea mai mare valoare mai mică sau egală cu acel element și care este o putere a lui 2. Șirul ***a*** și valoarea lui ***n*** sunt atât parametri de intrare, cât și de ieșire pentru subalgoritm.
Exemplu: dacă ***n*** = 4 și ***a*** = (3, 1, 24, 9), atunci ***a*** va deveni (3, 2, 1, 1, 24, 16, 9, 8), iar ***n*** = 8.

Subiectul II (15 puncte)

Se dă următorul subalgoritm care are ca parametri de intrare două numere naturale ***a*** și ***b*** ($0 < \boldsymbol{a} \leq 10000$, $0 \leq \boldsymbol{b} \leq 10000$).

```
Subalgoritm F(a, b):
      c ← 1
      CâtTimp b > 0 execută
         Dacă b mod 2 = 1 atunci        { mod calculează restul împărțirii întregi a lui b la 2 }
            c ← (c * a) mod 10
         SfDacă
         a ← (a * a) mod 10
         b ← b div 2                    { div calculează câtul împărțirii întregi a lui b la 2 }
      SfCâtTimp
      returnează c
SfSubalgoritm
```

a. Enunțați problema pe care o rezolvă subalgoritmul dat.
b. Ce valoare returnează apelul `F(1002,6)`?
c. Scrieți o variantă *recursivă* a subalgoritmului, respectând antetul subalgoritmului din varianta iterativă (ne-recursivă) de mai sus.

Subiectul III (25 puncte)

Insule

O companie aeriană a pus la dispoziția călătorilor șirul altitudinilor punctelor geografice deasupra cărora se va călători cu avionul de la Cluj-Napoca la New York. Șirul ***a*** are ***n*** ($3 \leq \boldsymbol{n} \leq 10\,000$) elemente numere naturale strict mai mici decât 30000, unde corespunzător punctelor de pe uscat altitudinile sunt numere nenule, iar corespunzător oceanului altitudinile sunt egale cu 0. Numim *insulă* o secvență de puncte de pe uscat mărginită de apă.

Scrieți un program care:

1. Determină și afișează numărul de ordine al punctului unde începe și numărul de ordine al punctului unde se termină cea mai lungă insulă. Dacă există mai multe soluții, se va determina o singură soluție. Dacă nu există nicio insulă, atunci se va afișa mesajul „*Nu există insule*”.
2. Stabilește dacă cea mai lungă insulă este de tip *munte* sau nu și afișează mesajul „*Este munte*”, respectiv „*Nu este munte*”. O insulă se consideră de tip *munte*, dacă altitudinile corespunzătoare ei formează un șir nevid strict crescător până la un moment dat, apoi formează un șir nevid strict descrescător.
3. Verifică dacă altitudinile corespunzătoare punctelor de pe uscat sunt *distincte* (două câte două) sau nu și afișează mesajul ”*Altitudinile sunt distincte*”, respectiv „*Altitudinile nu sunt distincte*”.
4. Dacă răspunsul la punctul 3 este „*nu*”, determină și afișează valoarea celei mai frecvente altitudini și numărul aparițiilor acesteia. Dacă există mai multe soluții, se va determina o singură soluție.

Exemplul* 1:** dacă ***n = 15 și ***a*** = (10, 2, 1, 0, 7, 0, 1, 2, 13, 5, 0, 0, 8, 5, 2), există 2 insule, iar insula cea mai lungă se află între altitudinile cu numerele de ordine 7 și 10. Insula cea mai lungă este de tip *munte*. Altitudinile nu au valori distincte, iar altitudinea cea mai frecventă are valoarea 2 și apare de 3 ori.
Exemplul* 2:** dacă ***n = 10 și ***a*** = (1, 2, 0, 1, 2, 13, 0, 0, 1, 2), există o singură insulă care se află între altitudinile cu numerele de ordine 4 și 6 și *nu* este de tip *munte*. Altitudinile nu au valori distincte, iar una dintre cele mai frecvente altitudini are valoarea 1 și apare de 3 ori.

Observație: În exemple indexarea șirului ***a*** începe de la 1.

Scrieți subprograme pentru:
a. a citi șirul ***a*** și lungimea lui de la tastatură;
b. a determina numărul de ordine al punctului unde începe și a numărul de ordine al punctului unde se termină cea mai lungă insulă;
c. a stabili proprietatea unei insule de a fi de tip *munte*;
d. a verifica dacă altitudinile corespunzătoare uscatului sunt distincte (două câte două) sau nu;
e. a determina cea mai frecventă altitudine de pe uscat și numărul de apariții ale acesteia.

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.