

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $a \div b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[a]$ partea întreagă a numărului real a .
- a. Scrieți ce se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 6, 16, 4273, 1679, 3165, 16, 200, 167. (6p.)
- b. Dacă primele două numere citite sunt, în această ordine, 2 și 22, scrieți o secvență de numere distincte din intervalul $[10^3, 10^4]$ care pot fi citite în continuare astfel încât, în urma executării algoritmului, să se afișeze valoarea 2. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat prima structură `cât timp...execută` cu o structură repetitivă de tip `pentru...execută`. (6p.)
2. Pentru a verifica dacă în tabloul unidimensional (7,8,12,15,17,20,22) există elementul cu valoarea x , număr natural, se aplică metoda căutării binare. Scrieți trei valori posibile ale lui x pentru ca succesiunea de elemente ale tabloului a căror valoare se compară cu valoarea lui x pe parcursul aplicării metodei indicate să fie 15, 20, 17. (6p.)
3. Două orașe participă la un studiu și pentru fiecare dintre acestea se memorează date obținute în urma recensământului populației: un cod al orașului (reprezentat ca o literă mare a alfabetului englez), numărul de locuitori (număr natural) și suprafața (număr real). Variabilele `cod1`, `nrLocuitori1` și `suprafata1` memorează datele obținute despre primul oraș, iar variabilele `cod2`, `nrLocuitori2` și `suprafata2` memorează datele obținute despre al doilea oraș. Densitatea unui oraș este un număr egal cu raportul dintre numărul de locuitori și suprafața acestuia. Declarați variabilele `cod1` și `cod2`, respectiv `suprafata1` și `suprafata2` și scrieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze pe ecran codul celui oraș, dintre cele două precizate mai sus, care are densitatea mai mare, sau mesajul `egale`, dacă cele două densități sunt egale. (6p.)

```
citește n, x (numere naturale nenule)
nr ← 0; i ← 1
cât timp i ≤ n execută
    citește y (număr natural)
    cât timp y ≠ 0 și y % 2 ≠ x % 2 execută
        y ← [y/10]
    dacă y = x atunci nr ← nr + 1
    i ← i + 1
scrie nr
```

SUBIECTUL al III-lea

(30 de puncte)

1. Se citesc trei numere naturale nenule, x , y și n ($x < y < n$), și se cere să se scrie cea mai mare valoare naturală din intervalul $[1, n]$ pentru care atât restul împărțirii la x , cât și restul împărțirii la y , sunt egale cu 2, sau 0 dacă nu există o astfel de valoare. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. Exemplu: pentru $x=10$, $y=101$ și $n=3000$, se scrie numărul 2022 (pentru numerele 2, 1012 și 2022 atât restul împărțirii la 10, cât și restul împărțirii la 101, este 2). (10p.)
2. Un șir de valori naturale reprezintă o listă a unor date codificate de identificare ale invitaților la o petrecere; fiecare invitat are un prenume și un nume, formate din câte un singur cuvânt. Cuvintele sunt codificate prin numere naturale nenule, iar codurile corespunzătoare prenumelui, respectiv numelui unei persoane apar în listă în această ordine, urmate de o valoare 0. Scrieți un program C/C++ care citește de la tastatură numere naturale, în această ordine: un număr n ($n \in [2, 100]$), cele $3 \cdot n$ elemente ale unui tablou unidimensional, numere din intervalul $[0, 10^4]$, reprezentând o listă de date de tipul precizat mai sus, apoi un număr nenul x . Programul afișează pe ecran, separate prin câte un spațiu, codificări ale **numelor** tuturor invitaților care au **prenumele** codificat prin x , ca în exemplu, sau mesajul `NU` dacă nu există astfel de invitați. Exemplu: pentru $n=4$, tabloul (10, 121, 0, 101, 512, 0, 10, 9876, 0, 2103, 10, 0) și $x=10$ se afișează pe ecran 121 9876 (10p.)
3. Se citește de la tastatură un număr natural, n ($n \in [1, 10^9]$), și se cere să se scrie în fișierul text `bac.txt` cel mai mare număr natural p cu proprietatea că numărul 9^p este divizor al numărului obținut prin evaluarea produsului $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate. Exemplu: dacă $n=14$, fișierul conține numărul 2 ($9^2=81$ este divizor al lui $1 \cdot 2 \cdot 3 \cdot \dots \cdot 14=87178291200$). a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.) b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)