



**SUBIECTUL al II-lea**

**(40 de puncte)**

1. **Algoritmul alăturat este reprezentat în pseudocod.**  
S-a notat cu  $a \% b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$  și cu  $[c]$  partea întreagă a numărului real  $c$ .
- a. Scrieți numărul afișat în urma executării algoritmului dacă se citește valoarea 2592. **(6p.)**
- b. Scrieți trei numere din intervalul  $[10^3, 10^4)$  care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 20. **(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 de tip `pentru...execută`. **(6p.)**
2. Pentru fiecare dintre cele trei specii de flori dintr-o florărie se memorează date specifice: un cod, reprezentând specia, și numărul de exemplare disponibile. Variabilele `cod1` și `nr1` memorează datele specifice pentru prima specie, variabilele `cod2` și `nr2` datele specifice pentru cea de a doua specie, iar variabilele `cod3` și `nr3` datele specifice pentru cea de a treia specie. Știind că cele trei specii au un număr diferit de exemplare, scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran codul speciei din care există cele mai puține exemplare. **(6p.)**
3. Pentru a verifica dacă în tabloul unidimensional  $(48, 24, 16, 14, 9, 8, 4)$  există elementul cu valoarea  $x=4$  se aplică metoda căutării binare. Scrieți succesiunea de elemente a căror valoare se compară cu  $x$  pe parcursul aplicării metodei. **(6p.)**

```

citește n (număr natural nenul)
nr ← 0
c ← 9
cât timp c ≥ 0 execută
| m ← n
| cât timp m ≠ 0 și m % 10 ≠ c execută
|   m ← [m / 10]
|   ■
|   dacă m ≠ 0 atunci
|     nr ← nr * 10 + m % 10
|     ■
|   c ← c - 1
|   ■
scrie nr

```

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Se citește un număr natural  $n$  ( $n \geq 1$ ) și se cere să se scrie cea mai mare valoare din intervalul  $[1, n]$ , cu proprietatea că este o putere a lui 2. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **Exemplu:** dacă  $n=20$ , se scrie 16. **(10p.)**
2. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $n \in [2, 10^2]$ ) și cele  $n$  elemente ale unui tablou unidimensional, numere naturale din intervalul  $[1, 10^9]$ . Programul transformă tabloul în memorie, eliminând numai ultimul element par, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, separate prin câte un spațiu, iar dacă nu există niciun element par, se afișează pe ecran mesajul `nu exista`. **Exemplu:** pentru  $n=8$  și tabloul  $(20, 25, 10, \underline{90}, 45, 163, 45, 3)$  se obține tabloul  $(20, 25, 10, 45, 163, 45, 3)$  **(10p.)**
3. Șirul de mai jos este definit alăturat:
- $$f_n = \begin{cases} n & \text{dacă } n \leq 10 \\ 2 \cdot f_{n-1} & \text{dacă } n > 10 \end{cases}$$
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 40, 80, 160....
- Se citește de la tastatură un număr natural  $x$  ( $x \in [1, 10^9]$ ), termen al șirului dat, și se cere să se scrie în fișierul `bac.out`, separați prin câte un spațiu, în ordine strict descrescătoare, toți termenii șirului mai mici sau egali cu  $x$ . Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare. **Exemplu:** dacă valoarea citită de la tastatură este 80 se scriu în fișier numerele 80 40 20 10 9 8 7 6 5 4 3 2 1
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**