

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 .
- a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește, în această ordine, numerele **21, 38 și 4**. (6p.)
- b. Dacă pentru m și x se citesc numerele **20**, respectiv **2020**, scrieți cea mai mică și cea mai mare valoare care pot fi citite pentru variabila n , astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze **2020**. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **repetă . . . până când** cu o structură de alt tip. (6p.)
2. Variabila p memorează, pentru fiecare dintre cele **20** de zone de parcare ale unui oraș, următoarele date specifice: identificatorul zonei, numărul de locuri închiriate pe parcursul lunii curente, precum și prețul practicat în zona respectivă pentru închirierea unui loc pentru o lună. Știind că expresiile C/C++ de mai jos au ca valori câte un număr natural, reprezentând identificatorul primei zone, respectiv suma obținută în urma închirierii pe parcursul lunii curente a tuturor locurilor de parcare din această zonă, scrieți definiția unei structuri cu eticheta **parcare**, care să permită memorarea datelor specifice unei zone de parcare, și declarați corespunzător variabila p .
p[0].id p[0].numar*p[0].pret (6p.)
3. Variabila i este de tip întreg, iar variabila a memorează un tablou bidimensional cu **5** linii și **7** coloane, numerotate începând cu **0**, cu elemente numere întregi. Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran, separate prin câte un spațiu, indicii liniilor cu proprietatea că primul sau ultimul lor element are valoarea **2020**. (6p.)

```

citește m, n, x
    (numere naturale nenule, m ≤ n)
s ← 0; pm ← 1; pn ← 1
repetă
    dacă m % x = 0 atunci
        s ← s + m; pm ← x
    ──
    dacă n % x = 0 și m ≠ n atunci
        s ← s + n; pn ← x
    ──
    m ← m + pm
    n ← n - pn
până când m > n
scrie s
    
```

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul **duplicare** are doi parametri:
- n , prin care primește un număr natural ($n \in [1, 10^4]$);
 - d , prin care furnizează numărul obținut prin duplicarea fiecărei cifre impare a lui n sau -1 dacă acesta nu are nicio cifră impară.
- Scrieți definiția completă a subprogramului.
Exemplu: dacă $n=2019$, după apel $d=201199$. (10p.)
2. Un text are cel mult **100** de caractere, iar cuvintele sale sunt formate numai din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un număr natural n ($n \in [1, 10^2]$), apoi un text de tipul precizat mai sus, și afișează pe ecran cuvintele acestuia, pe rânduri separate, astfel încât primele poziții să fie ocupate de mulțimea formată de cele care au cel puțin n litere, iar următoarele poziții, în continuarea acestora, să fie ocupate de mulțimea celorlalte cuvinte. Cuvintele din aceeași mulțime sunt afișate într-o ordine oarecare, iar dacă una dintre cele două mulțimi este vidă, se afișează pe ecran doar mesajul **nu exista**.
Exemplu: pentru $n=5$ și textul **e1 mergea tot spre aleea pietruita** datele afișate pot fi cele alăturate. (10p.)
- ```

mergea
aleea
pietruita
e1
tot
spre

```
3. Fișierul **numere.in** conține pe prima linie un număr natural  $n$  ( $n \in [2, 10^9]$ ), iar pe a doua linie un șir de cel mult  $10^9$  numere naturale din intervalul  $[1, n]$ . Numerele din șir sunt ordonate descrescător și sunt separate prin câte un spațiu. Se cere să se determine numărul valorilor naturale distincte din intervalul  $[1, n]$  care **NU** se găsesc în șirul menționat mai sus. Numărul determinat se afișează pe ecran. Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.  
**Exemplu:** dacă fișierul conține numerele  
**10**  
**8 8 8 5 3 3**  
se afișează pe ecran **7** (în șir nu se găsesc valorile **10 9 7 6 4 2 1**).  
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)