

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Considerăm un alfabet A format dintr-o mulțime finită de caractere. Un cuvânt este o secvență nevidă de caractere distincte din A . Lungimea unui cuvânt este numărul de caractere din care acesta este format. Un p -sistem ($p \geq 1$) este definit ca o mulțime S de cuvinte, toate de lungime p , având proprietățile: (P₁) oricare două cuvinte din S au exact un caracter comun și (P₂) orice caracter din alfabetul A apare în cel puțin un cuvânt din S .

Exemplu: Pentru $A = \{ 'a', 'b', 'c' \}$, mulțimea $S = \{ "ab", "ac" \}$ este un 2-sistem.

a) Scrieți un 3-sistem peste alfabetul $A = \{ 'a', 'b', 'c', 'd', 'e', 'f' \}$. **(2p.)**

b) Scrieți un program C/C++ care:

b1) Citește de la tastatură un întreg m , $1 \leq m \leq 26$, și construiește alfabetul A format din primele m caractere ale alfabetului $\{ 'a', 'b', \dots, 'z' \}$. **(1p.)**

b2) Citește de la tastatură un întreg $n \geq 1$ și o secvență de n șiruri de caractere. Se presupune că fiecare șir citit este format din caractere distincte din A (este un cuvânt). Nu este necesară validarea datelor citite. **(1p.)**

b3) Verifică dacă există un p astfel încât mulțimea de cuvinte citite reprezintă un p -sistem. În caz afirmativ, va fi afișat mesajul "DA", în caz contrar "NU". **(6p.)**

4. O rețea de comparatori de tip n ($n \geq 2$) și de dimensiune m ($m \geq 1$) este o secvență (c_1, c_2, \dots, c_m) în care fiecare element c_i , numit **comparator**, este o pereche de numere întregi (j, k) cu proprietatea $1 \leq j < k \leq n$.

Exemplu: Rețeaua $R = ((1, 2), (2, 3))$ are tipul 3 și dimensiunea 2.

Dacă a este un vector de n numere întregi și R este o rețea, notăm cu $R(a)$ vectorul obținut aplicând următoarele transformări lui a : pentru fiecare comparator $c_i = (j, k)$, $1 \leq i \leq m$ din R , în ordinea în care aceștia apar în rețea, dacă $a[j] > a[k]$ atunci în vectorul a interschimbăm valorile de la pozițiile j și k .

Exemplu: Pentru $R = ((1, 2), (2, 3))$ și $a = (30, 20, 10)$ avem $R(a) = (20, 10, 30)$.

a) Fie $R = ((2, 4), (1, 2), (3, 4), (2, 3))$ și $a = (40, 30, 20, 10)$. Calculați $R(a)$ și scrieți valorile intermediare ale vectorului a corespunzătoare transformărilor efectuate. **(2p.)**

b) Dați exemplu de o rețea R de tip 4 cu proprietatea că pentru orice vector a format din 4 numere întregi distincte, $R(a)$ va avea elementele ordonate crescător. Justificare. **(4p.)**

c) Scrieți o funcție C/C++ care primește ca parametri numerele naturale n și m , o matrice R cu m linii și 2 coloane, reprezentând o rețea de comparatori, și un vector a de n numere întregi. Funcția va calcula vectorul $R(a)$ și va returna valoarea 1 dacă acesta are elementele ordonate crescător. În caz contrar, va returna valoarea 0. **(4p.)**

SUBIECTUL al III-lea

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră toate șirurile de lungime $l \in \{1, 2, 3\}$ formate din litere din mulțimea $\{a, b, c, d, e\}$. Câte dintre aceste șiruri au un număr par de vocale? (a și e sunt vocale) **(4p.)**

A. 79

B. 80

C. 81

D. 78

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Se consideră funcția recursivă F de mai jos. Ce valoare va returna apelul $F(0, 63, 64)$? **(6p.)**

```
int F(int u, int v, int t) {
    int m = (u + v) / 2;
    if (u >= v) { return u; }
    else if (m * m > t) { return F(u, m - 1, t); }
    else if (m * m < t) { return F(m + 1, v, t); }
    else { return m; }
}
```

3. O casă are $n \times m$ camere ($n, m \in \mathbb{N}$, $n, m \geq 2$) și este reprezentată ca o matrice cu n linii (numerotate de la 0 la $n - 1$) și m coloane (numerotate de la 0 la $m - 1$). Camera de la linia i ($0 \leq i \leq n - 1$) și coloana j ($0 \leq j \leq m - 1$) este identificată prin perechea de numere (i, j) . Toate camerele, cu excepția celor situate pe prima linie ($i = 0$) și a celor situate pe prima coloană ($j = 0$), au câte un comutator. Acționarea comutatorului din camera (i, j) (unde $1 \leq i \leq n - 1$, $1 \leq j \leq m - 1$) conduce la următorul rezultat: în fiecare dintre camerele (i, j) , $(i - 1, j)$, $(i, j - 1)$, $(i - 1, j - 1)$ (camera cu comutatorul, cea de deasupra, cea de la stânga și cea de deasupra și la stânga), dacă lumina era aprinsă, ea se stinge, iar dacă era stinsă, se aprinde.

- a) Matricea de mai jos reprezintă starea luminilor din fiecare cameră dintr-o casă cu $n = 5$ linii și $m = 4$ coloane. Pozițiile $(0, 1)$, $(0, 2)$, $(1, 1)$, $(1, 3)$, $(2, 2)$, $(2, 3)$ conțin valoarea 1, reprezentând faptul că luminile sunt aprinse în camerele respective, iar celelalte poziții conțin valoarea 0, reprezentând faptul că luminile sunt stinse. Comutatoarele din care camere trebuie acționate pentru a stinge luminile din toate camerele? **(4p.)**

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

- b) Scrieți o funcție C/C++ care primește 5 parametri de intrare: o matrice A de 0 și de 1, reprezentând starea luminilor din camere, dimensiunile n ($n \geq 2$) și m ($m \geq 2$) ale acesteia, precum și două numere naturale i, j ($1 \leq i \leq n - 1$, $1 \leq j \leq m - 1$), reprezentând o cameră (i, j) cu comutator. Funcția trebuie să modifice matricea primită ca parametru, astfel încât aceasta să conțină starea luminilor după acționarea comutatorului din camera (i, j) . Nu este necesară validarea parametrilor de intrare. **(6p.)**
- c) Scrieți o funcție C/C++ care primește ca parametri de intrare o matrice A de 0 și de 1, reprezentând starea luminilor din camere, precum și dimensiunile n ($n \geq 2$) și m ($m \geq 2$) ale acesteia. Funcția trebuie să returneze 1 dacă există o mulțime de camere astfel încât, acționând comutatorul în fiecare cameră din mulțime o singură dată, să se obțină stingerea luminilor din toate camerele. Altfel, funcția va returna 0. În rezolvare, puteți utiliza funcția de la punctul b). Nu este necesară validarea parametrilor de intrare. Se va folosi un algoritm cât mai eficient din punct de vedere al timpului de executare; soluțiile de tip backtracking vor primi cel mult **2p.** **(10p.)**