

Test la INFORMATICĂ

Se acordă 10 puncte din oficiu. Timpul de lucru este de 3 ore.

SUBIECTUL I

(30 de puncte)

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

1. Care dintre expresiile C/C++ de mai jos este adevărată dacă și numai dacă valoarea variabilei întregi  $n$  este un număr natural de două cifre având cifra unităților egală cu succesorul cifrei zecilor? (5p.)

- A.  $(n > 10) \&\& (n < 100) \&\& ((n \% 10) + 1 == n / 10)$       B.  $(n > 10) \&\& (n < 90) \&\& ((n \% 10 - n / 10) == 1)$   
C.  $(n > 10) \&\& (n \% 10 - 1 == (n / 10) \% 10)$       D.  $(!(n \% 100)) \&\& (n > 10) \&\& (n / 10 == n \% 10 + 1)$

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

2. Se consideră subprogramul Z din dreapta, descris în pseudocod. Subprogramul primește la intrare, în parametrul  $n$ , un număr natural.

```
subprogram Z(n)
    (n - număr natural)
i ← 0
j ← 1
k ← 1
cât timp k ≤ n
    dacă j ≤ i atunci
        scrie '0'
        j ← j + 1
    altfel
        scrie '1'
        i ← i + 1
        j ← 1
    k ← k + 1
```

- a. Ce afișează apelul Z pentru parametrul  $n = 10$ ? (5p.)  
b. Care este valoarea minimă a parametrului de intrare  $n$  astfel încât  $Z(n)$  să afișeze 100 de valori de 1? (5p.)  
c. Câte valori de 0 afișează apelul  $Z(100)$ ? (5p.)  
d. Scrieți o funcție C/C++ care implementează subprogramul Z alăturat. (10p.)

SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Care este numărul matricilor pătrate, având 4 linii și 4 coloane, simetrice față de diagonala principală, ce conțin doar valorile 0 și 1? (5p.)

- A. 256      B. 512      C. 128      D. 1024

2. Fie  $P$  un poligon convex cu 10 vârfuri. Câte triunghiuri pot fi formate cu vârfurile lui  $P$ , ale căror laturi nu sunt laturi ale lui  $P$ ? (5p.)

- A. 30      B. 40      C. 50      D. 60

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

3. Considerăm o rețea socială formată din  $n \geq 2$  persoane, fiecare persoană având cel puțin  $k$  prieteni care fac parte din rețea ( $1 \leq k < n$ ).

- a) Descrieți în limbaj natural un algoritm care să partiționeze rețeaua în cel mult  $n - k$  grupuri, astfel încât fiecare persoană să aparțină unui singur grup, în care este în relație de prietenie cu toate celelalte persoane din grup. Justificați corectitudinea acestuia. (4p.)  
b) Presupunem că persoanele sunt identificate prin numere distincte din mulțimea  $\{0, \dots, n - 1\}$ . Scrieți un program C/C++ care citește de la tastatură rețeaua socială și numărul  $k$  și afișează grupurile de persoane descrise la punctul precedent. Nu este necesară validarea datelor de intrare. (6p.)  
4. Spunem că două grafuri  $G = (V, E)$  și  $G' = (V', E')$  sunt izomorfe dacă există o funcție bijectivă  $f : V \rightarrow V'$  astfel încât  $(u, v) \in E$  dacă și numai dacă  $(f(u), f(v)) \in E'$ , unde  $V, V'$  sunt mulțimile de noduri ale celor două grafuri, iar  $E, E'$  mulțimile de muchii.  
a) Arătați că grafurile  $G = (\{0, 1, 2, 3, 4, 5\}, \{(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 0)\})$  și  $G' = (\{0, 1, 2, 3, 4, 5\}, \{(0, 2), (2, 4), (4, 1), (1, 3), (3, 5), (5, 0)\})$  sunt izomorfe. (1p.)  
b) Arătați că grafurile  $G = (\{0, 1, 2, 3, 4\}, \{(0, 1), (1, 2), (2, 3), (3, 4)\})$  și  $G' = (\{0, 1, 2, 3, 4\}, \{(0, 1), (1, 2), (2, 3), (3, 4), (4, 0)\})$  nu sunt izomorfe. (1p.)

- c) Descrieți în limbaj natural un algoritm care verifică dacă două grafuri conexe, în care fiecare nod are maxim doi vecini, sunt izomorfe. Justificați corectitudinea acestuia. (3p.)
- d) Scrieți o funcție C/C++ care are ca parametri de intrare matricile de adiacență ale două grafuri conexe, în care fiecare nod are maxim doi vecini. Funcția va returna valoarea 1 dacă cele două grafuri sunt izomorfe, 0 în caz contrar. Nu este necesară validarea parametrilor de intrare. (5p.)

### SUBIECTUL al III-lea

(30 de puncte)

#### Scrieți pe foaia de examen răspunsul pentru itemul 1.

1. Fie funcțiile recursive  $f$  și  $g$  din dreapta, diferența dintre ele fiind cazul de bază. Ce valoare va returna apelul  $g(4)$ ? (5p.)

```
int f(int x) {
    if (x <= 1) return 1;
    return f(x - 1) + f(x - 2);
}
int g(int x) {
    if (x <= 1) return x;
    return g(x - 1) + g(x - 2);
}
```

Pentru itemul 2, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

2. Fie  $a \geq 2$  și  $b \geq 2$  două numere naturale. Care trebuie să fie relația dintre  $a$  și  $b$  pentru ca apelul  $f(a)$  să întoarcă aceeași valoare cu  $g(b)$ ? (5p.)

- A.  $a = b$       B.  $a = b + 1$       C.  $a = b - 1$       D.  $a = b + 2$

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

3. Simplificând, instrumentele de tip *model mare de limbaj* (*LLM, large language model*) generează text cuvânt cu cuvânt, alegând următorul cuvânt în mod probabilist, în funcție de unele dintre cuvintele generate anterior. Probabilitatea cu care se alege următorul cuvânt depinde de textul folosit în datele de antrenament. De exemplu, un *LLM* antrenat pe povești în limba română va genera cu probabilitate foarte mare cuvântul **NICIODATĂ** imediat după cuvintele **A FOST ODATĂ CA**.

În această problemă, vom construi un model de limbaj simplificat, care alege următorul cuvânt de generat în funcție de cele două cuvinte generate anterior. Vocabularul conține 50257 cuvinte distincte, fiecare cuvânt fiind reprezentat printr-un număr natural de la 0 la 50256, iar textul de antrenament este dat printr-un tablou  $V$  de lungime  $1 \leq N \leq 1000000$ , indexat de la 0 la  $N - 1$ . Pornind de la definițiile din dreapta, fără a valida datele de intrare, scrieți următoarele funcții C/C++:

```
const int MAXW = 50257;
const int MAXN = 1000000;
typedef struct pair
{
    int w1; // 0 <= w1 < MAXW
    int w2; // 0 <= w2 < MAXW
} pair;
```

- a) `int mostLikely(int *V, int N, pair P)` (4p.)  
Determină cuvântul ce apare cel mai frecvent în textul de antrenament imediat după perechea de cuvinte  $P$ . Dacă nu există un astfel de cuvânt sau dacă nu este unic, funcția întoarce un cuvânt oarecare, la alegerea programatorului.
- b) `int pairs(int *V, int N, pair *Pairs)` (4p.)  
Determină perechile distincte de cuvinte care apar unul imediat după altul în text. Funcția stochează perechile în ordine lexicografică în tabloul `Pairs` și întoarce  $L$ , numărul de perechi distincte. Tabloul `Pairs` este alocat înainte de apelul funcției și are  $N - 1$  elemente.
- c) `int pairIndex(pair *Pairs, int L, pair P)` (4p.)  
Determină, prin căutare binară, poziția perechii  $P$  în tabloul `Pairs`, care conține cele  $L$  perechi distincte calculate la punctul anterior. Întoarce  $-1$  dacă  $P$  nu apare în `Pairs`.
- d) `void learn(int *V, int N, pair *Pairs, int L, int *Next)` (4p.)  
Determină, pentru fiecare dintre perechile distincte de cuvinte, memorate în tabloul `Pairs` de lungime  $L$ , care este cuvântul întors de funcția `mostLikely` pentru acea pereche și păstrează rezultatele în parametrul de ieșire `Next`. Tabloul `Next` este alocat înainte de apelul funcției și are  $L$  elemente.
- e) `void generate(int *V, int N, int K, int *R, int X, int Y)` (4p.)  
Generează un text de  $K \geq 2$  cuvinte. Textul începe cu cuvintele  $X$  și  $Y$  și este stocat în parametrul de ieșire `R`. Tabloul `R` este alocat înainte de apelul funcției și conține  $K$  elemente. Cuvântul de la poziția  $i \geq 2$  este ales ca fiind cuvântul asociat perechii  $(R[i - 1], R[i - 2])$  de tabloul `Next`, dacă perechea apare în `Pairs`, sau în mod arbitrar, altfel. Calculați `Pairs` și `Next` prin apeluri la funcțiile de la subpunctele anterioare.