

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 .

- Scriveți ce se afișează dacă se citește numărul 10. (6p.)
- Scriveți două valori distincte care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, numărul de valori egale cu 1 afișate să fie 4. (6p.)
- Scriveți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scriveți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat prima structură repetitivă **cât timp...execută** cu o structură de tip **pentru...execută**. (6p.)

```

citește n
(număr natural,  $n \geq 2$ )
i ← 1
cât timp i ≤ n execută
    m ← i
    cât timp m % 2 = 0 execută
        m ← [m/2]
    dacă m ≠ i atunci
        scrie m, ' '
    i ← i + 1

```

- Un graf neorientat cu 6 noduri, numerotate de la 1 la 6, are muchiile [1,2], [2,3], [2,4], [2,5], [4,5], [4,6], [5,6]. Scrieți listele de adiacență ale unui graf parțial al său care să fie conex și fără cicluri. (6p.)
- Variabilele c și i sunt de tip întreg, iar variabila s permite memorarea unui șir de cel mult 20 de caractere. Se citesc de la tastatură 10 cuvinte, formate din litere mici ale alfabetului englez și separate prin Enter. Scrieți secvența de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila c să memoreze valoarea 1 dacă există printre cuvintele citite cel puțin unul format din două litere și care să conțină o vocală și o consoană, sau valoarea 0 altfel. Se consideră vocale literele a, e, i, o, u .

Exemplu: dacă se citesc cuvintele alăturate, variabila c are valoarea 1.

```

c = .....;
for(i=1; i<=10; i++) {  cin>>s;  |  scanf("%s", s);
                        .....
}

```

dacă
au
plecat
el
nu
primește
și
înghetata
de
fragi

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

- La un laborator sunt studiate aglomerările de fulgi de nea formate din câte **nouă** cristale de **patru** tipuri diferite date (notate cu 1, 2, 3 sau 4), astfel încât din fiecare tip să existe cel puțin câte un cristal. O astfel de aglomerare de fulgi a fost reprezentată printr-un număr natural, în care fiecare cifră reprezintă tipul unui cristal. Subprogramul **fulg** are un parametru, n , prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează valoarea 1, dacă prin n este reprezentată o aglomerare de fulgi de nea dintre cele studiate, sau 0 în caz contrar. Scrieți în C/C++ definiția completă a subprogramului.

Exemplu: dacă $n=112243413$ subprogramul returnează 1, iar dacă $n=12314$ sau $n=112253513$ sau $n=112243457$ sau $n=111122223$, subprogramul returnează 0. (10p.)

- Fiind date două numere naturale a și b , numim valoare **generată** de a și b un număr obținut din a prin alipirea la stânga sau la dreapta sa a cifrelor lui b , în ordinea în care apar în acesta.

Exemplu: dacă $a=123$ și $b=45$, se pot obține două numere generate de acestea: 12345 și 45123.

Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul [2, 50], m și n , și construiește în memorie un tablou bidimensional cu m linii, numerotate de la 1 la m , și n coloane, numerotate de la 1 la n , în care fiecare element este egal cu cea mai mică valoare generată de numerele de ordine ale liniei, respectiv coloanei pe care se află.

Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

Exemplu: pentru $m=5$ și $n=4$ se obține tabloul alăturat.

(10p.)

11	12	13	14
12	22	23	24
13	23	33	34
14	24	34	44
15	25	35	45

- Fișierul **numere.in** conține un șir de cel mult 10^6 numere naturale din intervalul [0, 99]. Numerele din fișier sunt separate prin câte un spațiu.

Se cere să se determine primul și ultimul număr din șir care conțin cea mai mare cifră ce apare în scrierea numerelor din fișier. Numerele determinate se afișează pe ecran, în ordinea apariției lor în șir, separate printr-un spațiu. Dacă nu există două astfel de numere pe poziții distincte, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu: dacă fișierul conține numerele 34 5 38 30 87 70 11 8 82 25 se afișează pe ecran 38 82, dacă fișierul conține numerele 34 5 38 30 87 70 11 8 38 25 se afișează pe ecran 38 38, iar dacă fișierul conține numerele 34 5 38 30 se afișează pe ecran **nu exista**.

- Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

- Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)

Examenul național de bacalaureat 2024

**Proba E. d)
INFORMATICĂ**

**BAREM DE EVALUARE ȘI DE NOTARE
(comun pentru limbajele C/C++ și Pascal)**

Varianța 4

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

- Se punctează orice modalitate de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit în barem. Nu se acordă fracțiuni de punct.
- Se acordă zece puncte din oficiu. Nota finală se calculează prin împărțirea la zece a punctajului total acordat pentru lucrare.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț (de exemplu tipuri întregi cu semn pentru memorarea numerelor naturale, dimensiune a tablourilor) este acceptată din punctul de vedere al corectitudinii programului, dacă acest lucru nu afectează funcționarea sa.

SUBIECTUL I (20 de puncte)

1a 2b 3d 4d 5c	5x4p.
----------------	-------

SUBIECTUL al II - lea (40 de puncte)

1.	a. Răspuns corect: 1 1 3 1 5	6p. Se acordă numai 1p. dacă doar un număr este conform cerinței pe poziția corespunzătoare, numai 2p. dacă doar două numere sunt conform cerinței pe poziția corespunzătoare, numai 3p. dacă doar trei numere sunt conform cerinței pe poziția corespunzătoare, numai 4p. dacă doar patru numere sunt conform cerinței pe poziția corespunzătoare și numai 5p. dacă în continuarea numerelor cerute, se precizează și alte numere.
	b. Pentru răspuns corect	6p. Se acordă câte 3p. pentru fiecare dintre cele două valori conform cerinței (numere din intervalul [16,31]).
	c. Pentru program corect -declaraire a variabilelor -citire a datelor -afișare a datelor -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. (*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței. 1p. 1p. 1p. 2p. 3p. 1p. 1p.
	d. Pentru algoritm pseudocod corect -utilizare a unei structuri repetitive de tipul indicat (*) -aspecte specifice ale secvenței obținute prin înlocuire, conform cerinței (**) -algoritm complet, corectitudine globală a algoritmului ¹⁾	6p. (*) Se acordă punctajul pentru orice formă de structură repetitivă de tipul cerut (pentru ... execută, for ... do etc.). 2p. (**) Se acordă câte 1p. pentru fiecare aspect specific 3p. (expresie de inițializare a contorului, expresie pentru valoarea finală a contorului, doar actualizare implicită a contorului) conform cerinței. 1p.
2.	Pentru răspuns corect	6p. Se acordă câte 2p. pentru fiecare aspect specific (liste de adiacență ale unui graf parțial, graf conex, graf fără cicluri) conform cerinței.
3.	Pentru răspuns corect -inițializare a variabilei c -verificare a proprietății cerute (*) -corectitudine globală a secvenței ¹⁾	6p. (*) Se acordă câte 1p. pentru fiecare aspect specific 1p. (identificare a lungimii unui cuvânt, identificare a unei 4p. vocale/consoane, identificare a unui cuvânt cu/fără 1p. proprietatea cerută, algoritm de bază de verificare a unei proprietăți într-o serie de valori) conform cerinței.

SUBIECTUL al III - lea

(30 de puncte)

1. Pentru subprogram corect -antet al subprogramului (*) -verificare a proprietății cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului¹⁾	10p. (*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametru de intrare) conform cerinței. 2p. 6p. (**) Se acordă câte 1p. pentru fiecare aspect specific 1p. (algoritm de bază pentru parcurgerea cifrelor unui număr, algoritm de bază pentru verificarea unei proprietăți/numărare pentru o serie de valori, număr care conține toate cele patru cifre indicate, număr care nu conține alte cifre decât cele indicate, număr de 9 cifre, cifre suport verificate) conform cerinței. 1p.
2. Pentru program corect -declarare a unei variabile care să permită memorarea unui tablou bidimensional, conform cerinței -citire a datelor -construire a tabloului cerut (*) -afișare a datelor în formatul cerut -declarare a variabilelor simple, corectitudine globală a programului¹⁾	10p. (*) Se acordă câte 1p. pentru fiecare aspect specific (acces la un element al tabloului, construire a unui număr generat prin alipirea unui număr de o cifră la o altă valoare, construire a unui număr generat prin alipirea unui număr de două cifre la o altă valoare, număr minim generat, elemente suport, construire în memorie) conform cerinței. 1p. 1p. 6p. 1p. 1p.
3. a. Pentru răspuns corect - descriere coerentă a algoritmului, conform cerinței (*) - elemente de eficiență justificate, conform cerinței b. Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorilor, conform cerinței (*),(**) - utilizare a unui algoritm eficient, conform cerinței (***) - declarare a variabilelor, afișare a datelor conform cerinței, corectitudine globală a programului¹⁾	2p. (*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. 1p. 1p. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. 8p. (***) Se acordă punctajul numai pentru un algoritm liniar (de complexitate $O(n)$), care utilizează eficient memoria. 1p. O soluție posibilă citește datele din fișier și, pe măsura parcurgerii lor determină cifra maximă întâlnită (cifMax), și primul și ultimul număr dintre cele citite care conțin această cifră (p, respectiv u). Pentru fiecare număr citit x se determină cifra maximă, cm (maximul dintre $x \% 10$ și $[x/10]$) și, dacă $cm > cifMax$ se actualizează cifMax, p și u ($cifMax \leftarrow cm$, $p \leftarrow x$, $u \leftarrow -1$), iar dacă $cm = cifMax$, se actualizează u ($u \leftarrow x$). După parcurgerea tuturor numerelor, se afișează p și u sau mesajul indicat (dacă $u = -1$). 5p. 1p. 1p.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.