

Examen de admitere Informatică, 17 Iulie 2021

Notă: Subprogramele pot fi scrise în limbajele de programare C/C++/ Pascal/Python. Proba de examen durează 3 ore.

Partea I (6p)

1. (0.75p) Un nufăr își dublează suprafața în fiecare zi.

- (a) (0.25p) În a 48-a zi nufărul ocupă toată suprafața unui lac. În a câta zi nufărul ocupă un sfert din suprafața lacului? (i) 46 (ii) 22 (iii) 12 (iv) 11
- (b) (0.5p) Considerăm un subprogram recursiv care primește ca parametri suprafața ocupată de nufăr azi, S (număr real pozitiv), și un număr natural X . Subprogramul returnează suprafața ocupată de nufăr în urmă cu X zile. Completați spațiile lipsă pentru condiția de oprire și apelul recursiv.

Subprogram $f(S, X)$
dacă ... (i) ... **returnează** S
altfel returnează ... (ii) ...

2. (1p) Se consideră că locurile dintr-un amfiteatru sunt reprezentate sub forma unei matrice de dimensiune $m \times n$ (unde $m, n \in \mathbb{N}$, $2 < m < 30$, $2 < n < 30$). Pentru participarea la un examen se dorește aranjarea candidaților după modelul de mai jos (pentru $m = 4$ și $n = 13$), unde 1 reprezintă loc ocupat, iar 0 reprezintă loc liber. Între oricare două locuri ocupate există un loc liber, între două rânduri ocupate există un rând liber.

```
1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0
```

- (a) (0.5p) Scrieți un subprogram care primește ca parametri numărul de rânduri și locuri pe rând și completează o matrice, declarată ca variabilă globală, cu structura specificată mai sus.
- (b) (0.25p) Care este numărul maxim de studenți care pot fi plasați într-un amfiteatru reprezentat sub forma unei matrice de dimensiune $n \times m$ ocupată după modelul de mai sus? ($\lfloor \rfloor$ reprezintă partea întreagă a unui număr real)
- (i) $\lfloor n/2 \rfloor \cdot \lfloor m/2 \rfloor$; (ii) $\lfloor (n+1)/2 \rfloor \cdot \lfloor (m+1)/2 \rfloor$; (iii) $m \cdot n/2$
- (c) (0.25p) Determinați numărul minim de locuri dintr-un amfiteatru ocupat la capacitate maximă cu 91 de studenți distribuiți în felul indicat mai sus.
- (i) 13×25 (ii) 13×26 (iii) 14×26 (iv) 14×25 (v) 12×25

3. (1.25p) Fie T un arbore cu rădăcină care are n noduri.

- (a) (0.25p) Câte muchii are arborele T ?
- (i) $n+1$ (ii) n (iii) $n-1$ (iv) $n-2$
- (b) (0.25p) În câte feluri se pot colora nodurile arborelui T cu roșu și negru astfel încât orice două noduri învecinate să aibă culori diferite?
- (i) 2 (ii) 1 (iii) n (iv) 2^n (v) $n-1$
- (c) (0.25p) Presupunem că arborele T are 9 noduri numerotate de la 1 la 9, rădăcina 1, și muchiile $[1, 2]$, $[1, 3]$, $[2, 4]$, $[2, 5]$, $[2, 6]$, $[2, 7]$, $[3, 8]$, $[8, 9]$. Care este numărul maxim de noduri roșii în arborele T dacă nodurile sunt colorate cu roșu și negru astfel încât oricare două noduri învecinate să aibă culori diferite?
- (d) (0.5p) Se consideră că un nod frunză este un nod care nu este tată al altui nod. Definiți un subprogram care primește ca parametri de intrare vectorul de tați al unui arbore cu rădăcina 1 și cu $n \leq 50$ noduri numerotate de la 1 la n , și numărul n , și returnează numărul frunzelor din arborele respectiv.

4. (1p) Se consideră un șir S_1 de n caractere ($n < 10000$). Un subșir, S_2 , al șirului S_1 este format dintr-o succesiune de caractere, nu neapărat consecutive ale șirului S_1 , în ordinea în care acestea apar în S_1 . Scrieți un subprogram care primește patru parametri: două șiruri S_1 și S_2 și lungimile lor, și verifică dacă S_2 este subșir al lui S_1 . În caz afirmativ, subprogramul afișează pozițiile caracterelor din șirul S_1 pe care se află elementele din S_2 . În cazul în care S_1 nu conține elementele lui S_2 se va afișa “Nu este subșir”.

De exemplu, pentru șirurile $S_1 = \text{“La ce examen primim rezultatul maine?”}$ și $S_2 = \text{“La mate?”}$ se va afișa

1 2 3 10 27 28 36 37

5. (1p) Un șir de biți este *k-special* dacă începe cu 1, conține bitul 1 de exact k ori, și între oricare doi de 1 consecutivi apare cel mult un 0. De exemplu, șirul 101100 este 3-special, dar șirul 110010 nu este. Fie A un tablou cu 0 și 1 care reprezintă un șir de n biți, $n \leq 100000$.

(a) (0.5p) Definiți un subprogram care primește ca parametru de intrare tabloul A , dimensiunea acestuia n și returnează **true** dacă tabloul A reprezintă un șir 6-special, și **false** în caz contrar.

(b) (0.5p) Câte șiruri 6-speciale de 12 biți există?

(i) 25 (ii) 32 (iii) 5 (iv) 6

6. (1p) Patru prieteni trebuie să treacă un pod șubred pe timp de noapte, înainte ca podul să fie luat de ape. Cei patru prieteni au diferite condiții fizice: Ana poate trece podul într-un minut, Bogdan are nevoie de 2 minute, Cristina de 5 minute, iar Dan de 10 minute. Podul nu poate susține mai mult de două persoane în același timp, iar aceste două persoane au nevoie de lanternă pentru a vedea unde pășesc. Drept urmare, după ce doi prieteni trec podul, lanterna trebuie adusă înapoi de cineva. Timpul în care trec doi prieteni este dat de timpul cel mai lung: dacă Ana și Dan trec în același timp, va dura 10 minute.

(a) (0.5p) Propuneți o ordine de trecere, astfel încât cei 4 prieteni să treacă podul în 17 minute.

(b) (0.5p) Scrieți un subprogram care primește ca parametri un tablou T , ($T_1 \leq T_2 \leq \dots \leq T_n$) reprezentând timpi de traversare, dimensiunea tabloului n ($n \leq 100$) și calculează timpul necesar trecerii tuturor celor n prieteni pentru următoarea strategie: prietenul cel mai rapid îi conduce pe fiecare, întorcându-se cu lanterna.

Partea a II-a (3p)

UEFA dorește realizarea unui sistem de înregistrare a rezultatelor meciurilor din grupele EURO 2020 pentru aflarea echipelor calificate în faza următoare. Se citește de la tastatură indicativul grupei ($A, B, \dots, F$), urmat de 6 rezultate de forma $a X Y$ b , semnificând că echipa a a marcat X goluri, iar echipa b a marcat Y goluri. Dacă o echipă câștigă meciul (are mai multe goluri înscrise), aceasta va acumula 3 puncte. Dacă scorul este egal, atunci fiecare echipă va acumula câte un punct.

Primele două clasate din fiecare grupă se vor califica direct în faza următoare. Dacă două echipe au același număr de puncte, vor fi departajate în funcție de golaverajul $G = GM - GP$, $GM =$ goluri marcate, $GP =$ goluri primite.

- (0.75p) Propuneți o structură de date pentru memorarea clasamentului, punctelor și golaverajului
- (0.75p) Scrieți un subprogram *citire_rezultate* care citește datele de intrare și care completează structura de date propusă.
- (0.75p) Scrieți un subprogram *ordonare_clasament* care ordonează descrescător echipele din fiecare grupă pe baza punctajului. Echipele cu același punctaj sunt ordonate descrescător după golaveraj.
- (0.75p) Scrieți un subprogram *afisare_calificate*, care folosește clasamentul ordonat anterior și afișează echipele calificate direct. Aceste echipe vor fi afișate în ordinea grupelor, adică $A_1, A_2, B_1, B_2, \dots$. Fiecare echipă va fi afișată pe o linie, alături de punctaj și golaveraj.

Exemplu:

A
a 0 3 b
c 1 1 d
a 0 2 c
b 3 0 d
b 1 0 c
d 3 1 a
B
...
Output:
Echipe calificate direct:
b 9 7
c 4 1
...

BAREM

Partea I

SUBIECTUL 1	0.75p
1a (i) 46	0.25p
1b (i) $X = 0$, (ii) $f(S/2, X - 1)$	0.5p

SUBIECTUL 2	1p
2a Variantă de rezolvare (în C++)	0.5p
<pre>int A[30][30]; void init(int m,int n) { for (int i=0;i<m;i++) for (int j=0;j<n;j++) if ((i%2==0)&&(j%2==0)) { A[i][j]=1; } else { A[i][j]=0; } }</pre>	
2b (ii) $\lfloor (n+1)/2 \rfloor \cdot \lfloor (m+1)/2 \rfloor$	0.25p
2c (i) $13 \times 25 = 325$	0.25p

SUBIECTUL 3	1.25p
3a (iii) $n - 1$	0.25p
Într-un arbore cu rădăcină, orice muchie este între un nod și părintele lui. Doar nodul rădăcină nu are părinte. Rezultă că numărul de muchii conicid cu numărul de noduri diferite de rădăcină, adică $n - 1$.	
3b (i) 2	0.25p

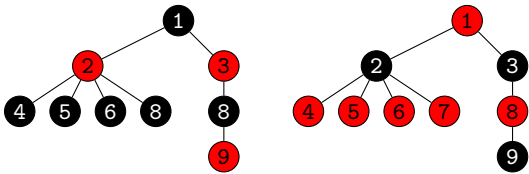
Fie $d(x)$ numărul de muchii de la rădăcina arborelui la nodul x din arbore, și c culoarea rădăcinii. Observăm că

- Toate nodurile cu $d(x)$ impar au culoarea diferită de c .
- Toate nodurile cu $d(x)$ par au culoarea c .

Deci, după ce am colorat rădăcina, avem o singură posibilitate să colorăm restul nodurilor din arbore. Rezultă că numărul de colorări cu roșu și negru al unui arbore coincide cu numărul de colorări posibile ale rădăcinii cu roșu sau negru, adică 2.

3c Răspuns: Numărul maxim de noduri roșii este 6, în a doua colorare.	0.25p
--	-------

Cele două colorări posibile sunt ilustrate mai jos:



3d Variantă de rezolvare (în C++)	0.5p
<pre>int nrFrunze(int* tata,int n) { int nrFii[51]; for (int i=1;i<=n;i++) nrFii[i]=0; for (int i=2;i<=n;i++) nrFii[tata[i]]++; int nrFrunze=0; for (int i=1;i<=n;i++) if (nrFii[i]==0) nrFrunze++; return nrFrunze; }</pre>	

SUBIECTUL 4	1p
-------------------	----

Variantă de rezolvare (în C++)

```

void subsir(char* S1, char* S2, int m, int n) {
    int p=0; // poziția curentă în șirul S2
    int subsir[n]; // pozițiile unde apare S2 în S1
    for (int i=0; i<m; i++) { // presupun că m este lungimea lui S1
        if (S1[i]==S2[p]) {
            subsir[p++]=i+1; // pentru că indexăm caracterele lui S1 începând de la 1
            if (p==n) {
                for (int j=0; j<n; j++) cout << subsir[j] << ' ';
                return;
            }
        }
    }
    cout<<"Nu este subsir";
}

```

SUBIECTUL 5 1p

5a Variantă de rezolvare (în C++)

0.5p

```

bool special6(int *A, int n) {
    int pos; // variabilă locală care reține poziția ultimului bit 1 găsit în A
    int n; // variabilă locală care reține numărul de biți 1 găsiți în A
    if (A[0]!=1) return false;
    else { n=1; pos=0; }
    for (int i=1; i<n; i++)
        if (A[i]==1) {
            n++;
            if (n>6 || (i-pos>2)) return false;
            pos=i;
        }
    return (n==6);
}

```

5b (ii) $2^5 = 32$ 0.5p

Orice șir 6-special de 12 biți este unic determinat de pozițiile $i_1, i_2, i_3, i_4, i_5, i_6$ unde apare 1 în șir. Primul bit este 1, deci $i_1 = 1$. Știm că $i_2 \in \{2, 3\}$, $i_3 \in \{i_2 + 1, i_2 + 2\}$, $i_4 \in \{i_3 + 1, i_3 + 2\}$, $i_5 \in \{i_4 + 1, i_4 + 2\}$ și $i_6 \in \{i_5 + 1, i_5 + 2\}$. Deci avem 2 posibilități pentru valoarea fiecărei poziții i_2, i_3, i_4, i_5, i_6 . Rezultă că sunt $2^5 = 32$ posibilități, adică 32 de șiruri 6-speciale de 12 biți.

SUBIECTUL 6 1p

6a Modul de traversare al podului este următorul: (0.5p)

```

Ana, Bogdan → (2)
← Ana (1)
Cristina, Dan → (10)
← Bogdan (2)
Ana, Bogdan → (2)
Total: 17

```

6b Orice implementare care efectuează calculul $T_1 \cdot (n - 2) + \sum_{i=2}^n T_i$ (0.5p)

Partea a II-a

În cele ce urmează se consideră că prima echipă se află pe poziția 0 în tabloul numit clasament.

1 0.75p

Varianta A: Folosirea unui tablou de structuri abstracte de date în care se rețin indicativul echipei, punctajul și golaverajul.

```

struct echipa {
    int indicativ, punctaj, golaveraj;
}
echipa clasament[24];

```

Varianta B: O matrice în care pe prima coloană se stochează indicativul echipei ($a = 0, b = 1, \dots$), pe a doua coloană numărul de puncte, iar pe a treia golaverajul.

```
int clasament[24][3];
```

2 0.75p

Citirea tuturor liniilor (6 indicative de grupă și 36 de meciuri) (0.25p), calcularea punctelor și golaverajului pe baza unui meci (0.25p), actualizarea clasamentului (0.25p)

3 0.75p

Parcursarea per grupe a clasamentului și ordonarea echipelor din fiecare grupă după punctaj și golaveraj folosind orice algoritm de sortare.

Variantă de rezolvare în C++:

```
void ordonare_clasament(echipa clasament[24]) {
    for(int gr = 0; gr < 6; gr++){
        int start = gr*4;
        int end = start+4;

        //ordonare între start si end
    }
}
```

4 0.75p

Parcursarea similar cu exemplul de la punctul 3, afișarea $clasament[start], clasament[start + 1]$

OBS: Orice variantă corectă de rezolvare va fi luată în considerare.