

Examenul de bacalaureat național 2019
Proba E. d)
Informatică
Limbajul C/C++

Varianta 5

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

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I (20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Indicați numărul de valori din intervalul $[10, 10^2)$ pe care le poate avea variabila x întregă x , astfel încât expresia C/C++ alăturată să aibă valoarea 1. $x \% 2 == x / 10$
- a. 1 b. 5 c. 11 d. 18
2. Subprogramul **f** este definit alăturat. Indicați apelul în urma executării căruia se afișează:
10011
- ```
void f (int n)
{ if (n!=0){ f(n/2);
 cout<<n%2; | printf("%d",n%2);
}
}
```
- a. **f(10);**                                      b. **f(19);**                                      c. **f(20);**                                      d. **f(2019);**
3. Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte 3 esențe distincte din mulțimea {**ambră, geranium, mosc, oud, violete**}. Două parfumuri sunt distincte dacă diferă prin cel puțin o esență. Primele patru soluții obținute sunt, în această ordine: (**ambră, geranium, mosc**), (**ambră, geranium, oud**), (**ambră, geranium, violete**) și (**ambră, mosc, oud**). Indicați succesiunea care **NU** se obține, prin această metodă, în ordinea dată.
- a. (**ambră, mosc, violete**)                                      b. (**mosc, oud, violete**)  
c. (**geranium, mosc, oud**)                                      d. (**oud, violete, geranium**)
4. Un graf neorientat are 7 noduri, numerotate de la 1 la 7, și muchiile [1,2], [2,3], [2,4], [3,4]. Indicați numărul muchiilor care trebuie adăugate pentru ca graful să devină complet.
- a. 17                                      b. 15                                      c. 4                                      d. 3
5. Într-un arbore cu 10 noduri, numerotate de la 1 la 10, gradul fiecărui nod este maximum 4. Dacă arborele este reprezentat prin vectorul de „tați” (**x,0,1,1,2,1,8,2,2,2**), numărul tuturor valorilor posibile ale lui **x** este:
- a. 3                                      b. 4                                      c. 5                                      d. 6

**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 dacă se citesc, în această ordine, numerele 15, 6, 4, 8, 3, 0. (6p.)
- b) Scrieți un set de 4 valori distincte care pot fi citite, în această ordine, astfel încât, în urma executării algoritmului, să se afișeze 10. (6p.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **pentru...execută** cu o structură repetitivă de alt tip. (6p.)

```
citește x
(număr natural nenul)
z ← 1
repetă
 citește y (număr natural)
 pentru d ← 2, y execută
 dacă x % d = 0 și y % d = 0 și d > z atunci
 z ← d
 x ← y
până când y = 0
scrie z
```

2. Expresiile de mai jos au ca valori un șir de maximum 30 de caractere, un număr natural, respectiv un număr real, reprezentând, în această ordine, următoarele informații memorate pentru o mașină: marca și date despre motor (turația medie și capacitatea cilindrică). Scrieți definiția unei structuri cu eticheta `masina`, care să permită memorarea informațiilor menționate pentru o mașină, și declarați corespunzător variabila `m`, de acest tip.

`m.marca          m.motor.turatie          m.motor.capacitate` (6p.)

3. Variabilele `a` și `b` permit memorarea câte unui șir de cel mult 20 de caractere. Scrieți ce se afișează în urma executării secvenței de mai jos.

```
strcpy(a,"succint"); strcpy(b,"cules");
a[4]='\0'; strcat(a,b+3);
cout<<a<<strlen(b); | printf("%s%d",a,strlen(b));
```

 (6p.)

### SUBIECTUL al III-lea

(30 de puncte)

1. Numim **sub-număr plat** al unui număr natural orice secvență formată din două cifre identice nenule, aflate pe poziții consecutive în scrierea acestuia.

Subprogramul `suma` are un singur parametru, `n`, prin care primește un număr natural ( $n \in [10, 10^9]$ ). Subprogramul returnează suma tuturor sub-numerelor plate ale lui `n`, sau valoarea 0 dacă nu există niciun astfel de sub-număr. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă `n=722722555`, atunci subprogramul returnează 154 ( $22+22+55+55=154$ ), iar dacă `n=140`, atunci subprogramul returnează 0. (10p.)

2. Suprafața unui parc este împărțită în  $n \cdot n$  zone de formă pătrată dispuse pe `n` rânduri și `n` coloane, numerotate de la 1 la `n`, fiecare zonă fiind destinată pentru a fi umplută cu pietriș sau pentru plantarea în aceasta fie doar a florilor, fie doar a ierbii, fie a unui singur copac. Zonelor li se atribuie câte un cod, în funcție de destinație: codul 0 pentru pietriș, codul 1 pentru flori, codul 2 pentru iarbă și codul 3 pentru copac.

Scrieți un program C/C++ care citește de la tastatură un număr natural, `n` ( $n \in [1, 10^2]$ ), și cele  $n \cdot n$  elemente ale unui tablou bidimensional, reprezentând codurile corespunzătoare zonelor din parc, în ordinea parcurgerii lor rând după rând, și a fiecărui rând de la stânga la dreapta.

Programul determină zonele destinate plantării unui copac, care au cel puțin o latură comună cu o zonă cu pietriș și, pentru fiecare dintre acestea, afișează pe ecran numărul de ordine al rândului pe care se află. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există nicio astfel de zonă, se afișează pe ecran mesajul **nu exista**.

**Exemplu:** pentru `n=6` și tabloul alăturat se afișează pe ecran, nu neapărat în această ordine, numerele: 1 2 4 4. (10p.)

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 2 | 2 | 2 | 3 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 3 | 3 | 1 | 1 |
| 3 | 1 | 3 | 3 | 1 | 3 |
| 1 | 0 | 2 | 2 | 1 | 1 |

3. Fișierul `bac.txt` conține un șir de cel mult  $10^6$  numere naturale din intervalul  $[0, 10^3)$ , separate prin câte un spațiu. Se cere să se afișeze pe ecran cea mai mare cifră dintre cele care sunt comune tuturor termenilor șirului aflat în fișier, sau mesajul **nu exista** dacă nu există nicio astfel de cifră. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele

`701 917 17 171 817 771 71`

atunci se afișează pe ecran 7 (cifrele 1 și 7 sunt comune tuturor termenilor șirului)

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

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