

Examenul național de bacalaureat 2025

Proba E. d)
INFORMATICĂ
Limbajul C/C++

Varianța 7

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

Filieră vocațională, profil militar, specializare matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei 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.

- Indicați valoarea expresiei C/C++ alăturate.
a. 14 b. 14.7 c. 15 d. 15.2
15.4/2+15/2
- Variabila **m**, declarată alăturat memorează, pentru un medicament codul, procentul de compensare și data expirării (ziua, luna și anul). Indicați o expresie C/C++ cu valoarea 1 dacă și numai dacă medicamentul expiră în anul 2025, după luna octombrie.
a. **m.data.an==2025 && m.data.luna>10**
b. **data.an.m==2025 && data.luna.m>10**
c. **m.expira.an==2025 && m.expira.luna>10**
d. **m.an.expira==2025 && m.luna.expira>10**

```
struct data  
{ int zi, luna, an; };  
struct medicament  
{ int cod;  
  float compensare;  
  data expira;  
};
```
- Un site de vânzări online oferă reduceri la încălțăminte pentru fiecare comandă care conține trei perechi de articole din categorii diferite din mulțimea ordonată {**balerini**, **botine**, **cizme**, **ghete**, **sandale**, **teniși**}, astfel încât o comandă să nu conțină simultan **ghete** și **cizme**, respectiv **balerini** și **sandale**. Două soluții diferă prin cel puțin o categorie. Utilizând metoda backtracking, se generează toate comenzile posibile, iar primele patru soluții generate sunt: (**balerini**, **botine**, **cizme**), (**balerini**, **botine**, **ghete**), (**balerini**, **botine**, **teniși**), (**balerini**, **cizme**, **teniși**). Indicați soluția generată imediat înainte de (**cizme**, **sandale**, **teniși**).
a. (**botine**, **ghete**, **sandale**) b. (**botine**, **cizme**, **sandale**)
c. (**botine**, **cizme**, **teniși**) d. (**botine**, **sandale**, **teniși**)
- Un arbore cu 6 noduri, numerotate de la 1 la 6, are muchiile [1,2], [1,3], [2,4], [2,5], [2,6]. Indicați două noduri care pot fi alese drept rădăcină, astfel încât arborele obținut, pentru fiecare dintre acestea, să aibă trei frunze.
a. 1,3 b. 2,4 c. 2,5 d. 4,6
- Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, are arcele (1,3), (2,1), (2,5), (2,6), (4,3), (6,4), (6,5). Indicați numărul minim de arce care trebuie adăugate pentru ca graful obținut să aibă două componente tare conexe.
a. 1 b. 2 c. 3 d. 4

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 valorile afișate în urma executării algoritmului, dacă se citește numărul 252. (6p.)
- Scriveți două numere distincte din intervalul $[10, 10^2]$ care pot fi citite, astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze două valori egale. (6p.)
- Scriveți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scriveți în pseudocod un algoritm echivalent cu cel dat, înlocuind a doua structură **cât timp...execută** cu o structură repetitivă cu test final. (6p.)

```

citește n
(număr natural,  $n \geq 2$ )
d ← 2; x ← 1; y ← 1
cât timp n > 1 execută
    dacă n % d = 0 atunci
        x ← d; y ← y * d
    cât timp n % d = 0 execută
        n ← [n / d]
    d ← d + 1
scrie x, ' ', y
    
```

- Subprogramul **f** este definit alăturat. Scrieți valoarea **f(2)**, respectiv valoarea **f(17)**. (6p.)

```

int f(int x)
{ if (x <= 4) return x;
  else return x - f(x - 2);
}
    
```

- Variabilele **s1** și **s2** permit accesarea câte unui șir de cel mult 50 de caractere, iar variabila **n** este de tip întreg. Scrieți șirul accesat prin variabila **s1**, precum și valoarea lui **n**, în urma executării secvenței alăturate. (6p.)

```

strcpy(s1, "parcarea");
strcpy(s2, strstr(s1, "car"));
n = strlen(s2);
strcpy(s1 + n - 2, s2 + n - 2);
    
```

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul consecutiv are doi parametri:

- n**, prin care primește un număr natural ($n \in [1, 10^4]$);
- f**, prin care furnizează un număr natural cu proprietatea: $(f-1) \cdot f < n \leq f \cdot (f+1)$.

Scrieți definiția completă a subprogramului C/C++.

Exemplu: dacă $n=19$ atunci $f=4$ ($3 \cdot 4 < 19 \leq 4 \cdot 5$).

(10p.)

- Parcarea unui mall are locuri de parcare dispuse pe **nr** rânduri, câte **np** pe fiecare rând, unul lângă altul. Trei prieteni vin cu câte o mașină la mall și caută un triplet de locuri libere alăturate, plasate toate trei doar pe primul sau toate trei doar pe ultimul rând al parării.

Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[3, 50]$, **nr** și **np**, reprezentând numărul de rânduri din parcare, respectiv numărul de locuri de pe fiecare rând, apoi **nr · np** valori din mulțimea $\{0, 1\}$, elemente ale unui tablou bidimensional cu **nr** linii și **np** coloane, reprezentând, în ordinea dispunerii lor pe rânduri, starea locurilor de parcare (0, pentru loc liber, sau 1, pentru loc ocupat).

Programul afișează pe ecran numărul de triplete de locuri libere pe care le pot găsi prietenii.

Exemplu: pentru **nr=4**, **np=5** și tabloul alăturat, se afișează pe ecran valoarea 3 (pe prima linie pot găsi tripletul format din al doilea, al treilea și al patrulea loc, iar pe ultima linie pot găsi tripletul format din primul, al doilea și al treilea loc sau tripletul format din al doilea, al treilea, și al patrulea loc).

1	0	0	0	1
0	0	0	1	0
0	1	0	0	0
0	0	0	0	1

(10p.)

- La o loterie se generează aleatoriu un șir de numere naturale și pentru fiecare număr generat, se inversează ordinea cifrelor. Dintre valorile distincte obținute se extrag trei numere, în această ordine: cel mai mic, cel mai mare dintre cele rămase, apoi cel mai mic dintre cele rămase.

Fișierul text **bac.in** conține cel mult 10^6 numere naturale din intervalul $[1001, 9999]$, cu cifra unităților nenulă, separate prin câte un spațiu, reprezentând termenii șirului generat aleatoriu în vederea extragerii.

Scrieți un program C/C++ care afișează pe ecran cele trei numere, în ordinea extragerii acestora. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există trei astfel de numere distincte, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

Exemplu: dacă fișierul conține numerele 1114 3212 3217 2855 7309 2131 2131 1238 7893 se afișează pe ecran, în această ordine, numerele 1312 9037 2123

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

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