

Examenul național de bacalaureat 2024

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

Simulare

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.

1. Indicați valoarea expresiei C/C++ alăturate. 6.3/20+24
- a. 0 b. 24 c. 24.315 d. 24.9
2. Subprogramul `f` este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.
`f(2024);`
- ```
void f(int x)
{ if(x>=24) if(x%2==0)
 { cout<<2; | printf("2");
 f(x/2);
 }
 else
 { f(x/2);
 cout<<4; | printf("4");
 }
 else cout<<0; | printf("0");
}
```
- a. 04424222                      b. 22220444                      c. 22242440                      d. 44402222
3. Utilizând metoda backtracking se generează toate variantele prin care patru persoane pot plăti o consumație totală de 24 de lei în următoarele condiții:
- fiecare plătește o sumă nenulă, strict mai mică decât 10;
  - primul plătește mai puțin decât al doilea, al doilea mai puțin decât al treilea, iar acesta mai puțin decât al patrulea.
- O soluție constă în patru valori, reprezentând, în ordine, sumele plătite de fiecare dintre cele patru persoane. Primele patru soluții generate sunt, în această ordine: (1, 6, 8, 9), (2, 5, 8, 9), (2, 6, 7, 9), (3, 4, 8, 9). Indicați soluția generată imediat după (3, 6, 7, 8).
- a. (3, 5, 7, 9)                      b. (4, 5, 6, 9)                      c. (4, 5, 7, 8)                      d. (5, 5, 6, 8)
4. În declararea de mai jos, câmpurile structurii memorează date referitoare la un joc video: categoria, numărul de niveluri de dificultate (maximum cinci) și numărul maxim de puncte care pot fi obținute la fiecare nivel.
- Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă numărul maxim de puncte care se pot obține pe nivelul notat cu 2 este strict mai mare decât 4.
- ```
struct joc
{ int categorie, nrNiveluri;
  int puncte[5];
};
```
- a. `joc(puncte[2])>4` b. `puncte.joc[2]>4` c. `puncte[2].g>4` d. `g.puncte[2]>4`
5. Într-un oraș sunt 5 parcuri, notate cu **A**, **B**, **C**, **D** și **E**, care sunt conectate prin piste pentru biciclete, pe care se circulă într-un singur sens, astfel: o pistă de la parcul **A** către parcul **B**, o pistă de la parcul **B** către parcul **E**, o pistă de la parcul **C** către parcul **D**, o pistă de la parcul **D** către parcul **C**, o pistă de la parcul **E** către parcul **A**, o pistă de la parcul **E** către parcul **C** și o pistă de la parcul **E** către parcul **D**. Indicați numărul minim de piste pentru care trebuie schimbat sensul de circulație, astfel încât graful orientat obținut, în care parcurile reprezintă vârfurile, iar pisteles reprezintă arcele, să fie tare conex.
- a. 1 b. 2 c. 4 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 și cu $[c]$ partea întregă a numărului real c .

- Scriveți valoarea afișată în urma executării algoritmului dacă se citește numărul 32625. (6p.)
- Scriveți două numere din intervalul $[10, 100]$ care pot fi citite pentru variabila n , astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea 2. (6p.)
- Scriveți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scriveți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat a doua structură **cât timp...execută** cu o structură repetitivă cu test final. (6p.)

```

citește n
  (număr natural nenul)
s ← 0; d ← 2
cât timp d*d ≤ n execută
  p ← 0
  cât timp n%d = 0 execută
    n ← [n/d]; p ← 1
  s ← s+d*p; d ← d+1
dacă n ≠ 1 atunci
  s ← s+n
scrie s
    
```

- Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați” $(8, 1, 4, 5, 0, 9, 4, 5, 4, 3)$. Enumerați toate nodurile arborelui care au câte un singur descendent direct („fiu”). (6p.)
- Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate începând de la 0, având inițial toate elementele egale cu 1.

Scriveți secvența de instrucțiuni C/C++ de mai jos, înlocuind punctele de suspensie cu instrucțiuni adecvate, dintre care **cel mult patru de atribuire**, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat.

```

for (i=0; i<5; i++)
  for (j=0; j<5; j++)
    .....
    
```

```

0 0 0 0 0
0 2 2 2 2
0 2 4 4 4
0 2 4 6 6
0 2 4 6 8
    
```

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

- Subprogramul **Impare** are un parametru, n , prin care primește un număr natural ($n \in [1, 10^4]$). Subprogramul returnează numărul obținut prin duplicarea fiecărei cifre impare a lui n sau -1 dacă acesta nu are nicio cifră impară. Scrieți definiția completă a subprogramului.

Exemplu: dacă $n=3361$, subprogramul returnează 3333611.

(10p.)

- Un **șablon** este un text în care cuvintele sunt separate prin câte un spațiu și sunt formate fie numai din litere mici și mari ale alfabetului englez, fie numai din caractere $*$, în ultimul caz numindu-se **cuvinte generice**. Lungimea unui cuvânt este egală cu numărul de caractere care îl compun.

Un computer generează o frază pe baza unui astfel de șablon, prin înlocuirea fiecărui cuvânt generic cu unul dintre cuvintele de aceeași lungime, preluat dintr-o listă dată.

Scriveți un program C/C++ care citește de la tastatură un număr natural, n ($n \in [1, 100]$), și o listă de n cuvinte, urmată de un șablon de tipul precizat. Fiecare cuvânt din listă este format din maximum 10 litere mici și mari ale alfabetului englez și la citire este introdus singur pe linie. Șablonul conține maximum 100 de caractere. Programul obține în memorie și apoi afișează pe ecran una dintre frazele care pot fi generate pe baza șablonului și a listei citite sau mesajul **imposibil** dacă nu se poate genera o astfel de frază.

Exemplu: dacă $n=6$, iar lista de cuvinte este cea alăturată,

```

pentru șablonul      Era o vreme ***** si ***** din belsug *****
se generează fraza   Era o vreme placuta si soare din belsug soare
sau fraza            Era o vreme calduta si soare din belsug acasa
etc., iar pentru șablonul *** o vreme ***** si *****
se afișează mesajul imposibil
    
```

```

rece
placuta
acasa
calduta
innorata
soare
    
```

(10p.)

- Un șir se numește de tip **api** dacă numărul de apariții ale fiecărui termen este mai mic sau egal cu acel termen și are o paritate egală cu a acestuia.

Fișierul **bac.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[1, 10^3]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, dacă șirul este de tip api, sau mesajul **NU** în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul conține numerele 6 27 2 6 27 6 6 14 14 2 27 se afișează pe ecran **DA**

(termenul par 6 apare de 4 ori, 4 fiind tot număr par și $4 \leq 6$, termenii pari 2 și 14 apar de câte 2 ori, 2 fiind tot număr par și $2 \leq 2$, respectiv $2 \leq 14$, iar termenul impar 27 apare de 3 ori, 3 fiind tot număr impar și $3 \leq 27$).

- Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
- Scriveți programul C/C++ corespunzător algoritmului proiectat. (8p.)