

Examenul național de bacalaureat 2021

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

Testul 9

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 muchie are extremități distincte și oricare două 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.

- Variabila x este de tip întreg și memorează un număr natural de cel puțin două cifre. Indicați o instrucțiune C/C++ în urma executării căreia se memorează în variabila x numărul obținut din valoarea sa inițială, prin inserarea cifrei 4 între cifra zecilor și cifra unităților.
 - $x = (x/10*10+4) *10+x\%10$;
 - $x = x/10+4+x\%10$;
 - $x = (x\%10*10+4) *10+x/10$;
 - $x = (x/10+4) *10+x\%10$;
- Utilizând metoda backtracking se generează toate posibilitățile de a planta de-a lungul unei străzi cinci arbori din mulțimea {salcie, carpen, larice, fag, ulm}. Două soluții sunt diferite dacă ordinea arborilor diferă. Primele patru soluții obținute sunt, în această ordine: (salcie, carpen, larice, fag, ulm), (salcie, carpen, larice, ulm, fag), (salcie, carpen, fag, larice, ulm), (salcie, carpen, fag, ulm, larice). Indicați antepenultima soluție generată.
 - (ulm, fag, larice, salcie, carpen)
 - (ulm, salcie, larice, fag, carpen)
 - (ulm, fag, carpen, larice, salcie)
 - (ulm, fag, carpen, salcie, larice)
- Indicați o declarație corectă pentru o variabilă x care să memoreze simultan tensiunea electrică (număr întreg) și intensitatea curentului electric (număr real) într-un circuit.
 - ```
struct circuit
{ int x.tensiune;
 double x.intensitate;
};
```
  - ```
struct x
{ int tensiune;
  double intensitate;
};
```
 - ```
int x.tensiune;
double x.intensitate;
```
  - ```
struct
{ int tensiune;
  double intensitate;
} x;
```
- Într-un arbore cu rădăcină un nod se află pe nivelul x dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea x . Pe nivelul 0 se află un singur nod (rădăcina). Un arbore are 8 noduri, numerotate de la 1 la 8, și muchiile [1,3], [1,7], [1,8], [2,4], [3,5], [3,6], [4,5]. Indicați două noduri care pot fi alese drept rădăcină astfel încât, pentru fiecare dintre arborii obținuți, pe nivelul 2 să fie un număr maxim de noduri.
 - 3,5
 - 4,7
 - 5,7
 - 6,8
- Un graf neorientat cu 5 noduri, numerotate de la 1 la 5, are muchiile [2,5], [3,1], [5,3], [5,4]. Indicați numărul minim de muchii care pot fi adăugate, astfel încât în graful obținut să existe cel puțin un ciclu elementar care să conțină toate nodurile acestuia.
 - 1
 - 2
 - 3
 - 4

SUBIECTUL al II-lea

(40 de puncte)

1. **Algoritmul alăturat este reprezentat în pseudocod.**
S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural, nenul, y .
- a. Scrieți ce se va afișa în urma executării algoritmului dacă se citește numărul 3. **(6p.)**
- b. Scrieți toate valorile care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, numărul total de caractere + (plus) afișate să fie cuprins în intervalul închis $[10, 20]$. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat doar una dintre structurile **pentru...execută** cu o structură repetitivă de tip **cât timp...execută**. **(6p.)**
2. Subprogramul f este definit alăturat. Scrieți toate valorile naturale din intervalul $[1, 10]$ pe care le poate avea x , astfel încât valoarea lui $f(10, x)$ să fie un număr strict mai mare decât 20. **(6p.)**
3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 4 linii și 5 coloane, numerotate începând de la 0, cu elemente numere întregi, inițial toate nule. Fără a utiliza alte variabile decât cele menționate, scrieți secvența de mai jos astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat.
- ```
for(i=0;i<4;i++)
 for(j=0;j<5;j++)

```
- (6p.)**

```

citește n
(număr natural nenul)
pentru i ← 1, n execută
 pentru j ← i, n execută
 scrie '+'
 dacă i%2 ≠ 0 atunci
 scrie '@'
```

```

int f(int a, int b)
{
 if(a > b) return a/b + f(a-b, b);
 if(a < b) return b/a + f(a, b-a);
 return 1;
}
```

```

0 1 2 0 1
1 2 0 1 2
2 0 1 2 0
0 1 2 0 1
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul `divizor` are patru parametri:
- $a, b$  și  $k$ , prin care primește câte un număr natural ( $a \in [0, 10^9], b \in [a, 10^9], k \in [1, 9]$ );
  - $nr$ , prin care furnizează numărul de valori naturale din intervalul  $[a, b]$  care sunt divizibile cu  $k$  și au ultima cifră egală cu  $k$ . Scrieți definiția completă a subprogramului.
- Exemplu:** dacă  $a=3, b=50$  și  $k=4$ , în urma apelului,  $nr=3$  (pentru numerele 4, 24, 44). **(10p.)**
2. Într-un text cu cel mult  $10^2$  caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și afișează pe ecran, pe linii separate, toate cuvintele sale care conțin o singură vocală distinctă, ca în exemplu. Dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele din mulțimea  $a, e, i, o, u$ .  
**Exemplu:** pentru textul **a plantat cinci lalele visinii sau rosii** se afișează pe ecran, nu neapărat în această ordine, cuvintele alăturate. **(10p.)**
3. Fișierul `numere.txt` conține cel mult  $10^5$  numere naturale din intervalul  $[1, 10^9]$ , câte unul pe fiecare linie. Se cere să se afișeze pe ecran cel mai mare număr care se poate forma cu toate cifrele care apar în numerele din fișier, ca în exemplu.
- Proiectați un algoritm eficient din punctul de vedere al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul alăturat, se afișează
- ```
9988887766333220
```
- (2p.)**
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

```

263
39628
79
887308
```