

Examenul de bacalaureat național 2020
Proba E. d)
Informatică
Limbajul C/C++

Testul 8

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.

- Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă x aparține reuniunii de intervale $[-2020, -2019] \cup [2019, 2020]$.
 - $x \geq -2020 \ \&\& \ x \leq -2019 \ \&\& \ x \geq 2019 \ \&\& \ x \leq 2020$
 - $!(x < -2020 \ || \ x > -2019) \ || \ !(x < 2019 \ || \ x > 2020)$
 - $x \geq -2020 \ || \ x \leq -2019 \ || \ x \geq 2019 \ || \ x \leq 2020$
 - $!(x < -2020 \ \&\& \ x > 2020 \ \&\& \ x > -2019 \ || \ x < 2019)$
- Utilizând metoda backtracking se generează toate valorile naturale formate cu cifre ale numărului 578 și care sunt strict mai mici decât acesta. Primele șase numere generate sunt, în această ordine, 5, 55, 555, 557, 558, 57. Indicați al câtelea număr generat este 7.
 - al 10-lea
 - al 9-lea
 - al 8-lea
 - al 7-lea
- Variabila e este declarată alăturat. Indicați o instrucțiune de atribuire corectă din punctul de vedere sintactic.

```
struct elev
{
    int cod;
    char initiala;
    struct
    {
        int nota1, nota2;
    } note;
}e;
```

 - $e(\text{initiala}) = 'A'$;
 - $e = (2020, 'A', 10, 9)$;
 - $e.\text{cod} = e.\text{initiala} - 'A'$;
 - $e.\text{elev}.\text{cod} = 2020$;
- Subprogramele $f1$ și $f2$ sunt definite mai jos.

```
int f1(int x, int y)
{
    if(y==0) return x;
    else return f1(x*y, y);
}

int f2(int x, int y)
{
    if(x==y) return x;
    else if(x>y) return f2(x-y, y);
    else return f2(x, y-x);
}
```

La apel, returnează valoarea celui mai mare divizor comun al celor două numere naturale nenule primite ca parametri:
 - atât $f1$ cât și $f2$
 - nici $f1$ nici $f2$
 - numai $f1$
 - numai $f2$
- Un graf orientat cu 5 vârfuri este reprezentat prin matricea de adiacență alăturată. Indicați numărul de vârfuri ale unui subgraf al acestuia care are un număr maxim de vârfuri izolate.

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

 - 1
 - 2
 - 3
 - 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 întregă a numărului real c .
- a. Scrieți valoarea afișată dacă se citește, în această ordine, numerele 201920 și 20. **(6p.)**
- b. Dacă numărul citit pentru y este 0, scrieți două numere din intervalul $[10^2, 10^3)$ care pot fi citite pentru x , astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, numărul afișat să fie 9. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind structura **repetă . . . până când** cu o structură repetitivă cu test inițial. **(6p.)**
- ```

citește x,y
(numere naturale)
repetă
 c ← x%10
 x ← [x/10]
 dacă c ≠ 0 atunci
 dacă y%10 < c atunci
 y ← y*10+c
 altfel
 y ← y*10-c
 până când x=0
scrie y

```
2. Un graf neorientat cu 6 noduri, numerotate de la 1 la 6, este reprezentat prin listele de adiacență alăturate. Pentru acest graf scrieți un lanț elementar, precum și un ciclu care nu este elementar. **(6p.)**
- 1: 2, 6  
2: 1, 4, 6  
3: 5, 6  
4: 2, 5  
5: 3, 4, 6  
6: 1, 2, 3, 5
3. Variabila  $i$  este de tip întreg, iar variabila  $s$  poate memora un șir de cel mult 20 de caractere. Scrieți ce se afișează în urma executării secvenței alăturate. **(6p.)**
- ```

strcpy(s, "stilou"+4);
cout<<s<<endl; | printf("%s\n", s);
strncpy(s, "stilou", 4); s[4]='\0';
for(i=0; i<4; i++)
  if(i%2==0) s[i]=s[0]+i-1;
  else s[i]=s[0]+3*(2*i/3-1);
cout<<s; | printf("%s", s);
  
```

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul **suma** are un singur parametru, n , prin care primește un număr natural ($n \in [2, 10^9]$). Subprogramul returnează suma divizorilor primi ai lui n . Scrieți definiția completă a subprogramului.
Exemplu: pentru $n=12$ subprogramul returnează 5 (divizorii primi ai lui 12 sunt 2 și 3). **(10p.)**
2. Scrieți un program C/C++ care citește de la tastatură numere naturale: n ($n \in [2, 20]$), k ($k \in [1, n]$) și $n \cdot n$ numere din intervalul $[0, 10^9]$, elemente ale unui tablou bidimensional cu n linii și n coloane, numerotate începând de la 1. Programul transformă tabloul în memorie, deplasând circular spre stânga, cu câte o poziție, toate elementele situate pe linia k , în stânga diagonalei principale, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, fiecare linie pe câte o linie a ecranului, cu elementele fiecărei linii separate prin câte un spațiu.
Exemplu: pentru $n=6$, $k=4$ și tabloul
- | | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | 7 | 8 | 9 | 0 | 1 |
| 2 | 4 | 6 | 8 | 0 | 2 |
| 1 | 3 | 5 | 7 | 9 | 5 |
| 6 | 2 | 7 | 4 | 5 | 7 |
| 3 | 9 | 4 | 8 | 5 | 7 |
- se obține tabloul
- | | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | 7 | 8 | 9 | 0 | 1 |
| 2 | 4 | 6 | 8 | 0 | 2 |
| 3 | 5 | 1 | 7 | 9 | 5 |
| 6 | 2 | 7 | 4 | 5 | 7 |
| 3 | 9 | 4 | 8 | 5 | 7 |
- (10p.)**
3. Fișierul **bac.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran pozițiile din șir pe care se află termeni precedați de un număr maxim de valori care au cifra unităților egală cu cifra unităților lor. Numerele afișate sunt separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
Exemplu: dacă fișierul **bac.in** conține numerele 112 12 5 25 88 15 2 19 32 179 35 621 pe ecran se afișează numerele de mai jos (termenii 32, respectiv 35 respectă proprietatea cerută):
9 11
- a. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**