

Examenul național de bacalaureat 2021
Proba E. d)
Informatică
Limbajul C/C++

Testul 1

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 o expresie C/C++ echivalentă cu cea alăturată. $(x > 5) \ \&\& \ (x < 20) \ || \ (x != y)$
- a. $(x > 5 \ || \ x < 20) \ \&\& \ (x == y)$ b. $!(x <= 5 \ || \ x >= 20) \ || \ (x != y)$
- c. $(x > 5 \ || \ x < 20) \ \&\& \ (x != y)$ d. $!(x < 5 \ || \ x > 20) \ \&\& \ (x != y)$
2. Subprogramul `f` este definit alăturat. Indicați valoarea `f(4770777, 7)`.
- ```
int f (int n, int k)
{
 if (n!=0)
 if (n%10==k) return 1+f(n/10,k);
 return 0;
}
```
- a. 2                      b. 3                      c. 4                      d. 5
3. Variabila `x` este declarată alăturat. Indicați secvența care, în urma executării, memorează în variabila `x` ziua, luna și anul corespunzătoare unei date calendaristice citite de la tastatură.
- ```
struct data
{
    int zi, luna, an;
}x;
```
- a. `cin>>x.zi>>x.luna>>x.an; | scanf("%d%d%d", &x.zi, &x.luna, &x.an);`
- b. `cin>>zi.x>>luna.x>>an.x; | scanf("%d%d%d", &zi.x, &luna.x, &an.x);`
- c. `cin>>x(zi, luna, an); | scanf("%d%d%d", &x(zi, luna, an));`
- d. `cin>>x(zi)>>x(luna)>>x(an); | scanf("%d%d%d", &x(zi), &x(luna), &x(an));`
4. Un arbore cu rădăcină are 8 noduri, numerotate de la 1 la 8, și muchiile $[1, 3]$, $[1, 7]$, $[1, 8]$, $[2, 4]$, $[3, 5]$, $[3, 6]$, $[4, 5]$. Știind că rădăcina arborelui este nodul numerotat cu 6, indicați nodurile de tip frunză ale arborelui dat.
- a. 6, 8 b. 2, 6 c. 4, 7, 8 d. 2, 7, 8
5. Un graf neorientat are 6 noduri și fiecare dintre acestea are gradul egal cu 1. Indicați numărul de componente conexe ale grafului.
- 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 .
- a. Scrieți numărul afișat în urma executării algoritmului dacă pentru n se citește valoarea 205579. **(6p.)**
- b. Scrieți trei numere din intervalul $[10^3, 10^4)$ care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 7. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura `repetă...până când` cu o structură repetitivă de alt tip. **(6p.)**
- ```

citește n (număr natural)
m ← 10
dacă n=0 atunci
 m ← 0
altfel
 repetă
 c ← n%10; n ← [n/10]
 dacă c ≤ m atunci m ← c
 altfel m ← -1
 până când n=0
scrie m

```
2. Utilizând metoda backtracking, sunt generate toate numerele din intervalul  $[100, 999]$ , cu proprietatea că au cifrele în ordine crescătoare, cifrele aflate pe poziții consecutive sunt de paritate diferită, iar suma lor este egală cu 14. Scrieți toate numerele generate, în ordinea obținerii lor. **(6p.)**
3. Variabilele  $s1$  și  $s2$  pot memora câte un șir de cel mult 50 de caractere. Scrieți ce se afișează în urma executării secvenței alăturate. **(6p.)**
- ```

strcpy(s1, "bac2021");
cout << strlen(s1) << endl; | printf("%d\n", length(s1));
strcpy(s2, s1+3); strcpy(s2+2, "20-");
strcat(s2, s1+3);
cout << s2; | printf("%s", s2);
    
```

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul `divX` are doi parametri, n și x , prin care primește câte un număr natural din intervalul $[2, 50]$. Subprogramul afișează pe ecran, în ordine descrescătoare, separate prin câte un spațiu, primele n numere naturale nenule divizibile cu x .
Scrieți definiția completă a subprogramului.
Exemplu: dacă $n=4$ și $x=15$ în urma apelului se afișează numerele 60 45 30 15 **(10p.)**
2. Scrieți un program C/C++ care citește de la tastatură numărul natural n ($n \in [2, 10^2]$) și elementele unui tablou bidimensional cu n linii și n coloane, numere naturale din intervalul $[0, 10^9]$.
Programul afișează pe ecran, separate prin câte un spațiu, elementele primului pătrat concentric, parcurs în sens invers al acelor de ceasornic, începând din colțul său stângasus, ca în exemplu. Primul pătrat concentric este format din prima și ultima linie, prima și ultima coloană a tabloului.
Exemplu: pentru $n=5$ și tabloul alăturat, se afișează pe ecran numerele
1 2 3 4 5 6 7 8 9 0 2 4 6 8 1 3 **(10p.)**
- | | | | | |
|---|---|---|---|---|
| 1 | 3 | 1 | 8 | 6 |
| 2 | 9 | 2 | 7 | 4 |
| 3 | 5 | 8 | 5 | 2 |
| 4 | 1 | 6 | 3 | 0 |
| 5 | 6 | 7 | 8 | 9 |
3. Fișierul `bac.in` conține cel mult 10^6 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran, în ordine descrescătoare, cele mai mari două numere de două cifre distincte care **NU** se află în fișier. Numerele afișate sunt separate printr-un spațiu, iar dacă nu există două astfel de numere, se afișează pe ecran mesajul `nu exista`. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
Exemplu: dacă fișierul `bac.in` conține numerele 12 235 123 67 98 6 96 94 123 67 98 100 se afișează pe ecran, în această ordine, numerele 97 95.
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**