

Examenul de bacalaureat național 2020
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 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 natural memorat în variabila întreagă n este divizibil cu 2 și cu 5.
 - $!(n\%2==1 \ || \ n\%5!=0)$
 - $!(n/2==1 \ \&\& \ n/5!=0)$
 - $n\%2==0 \ || \ !(n\%5==0)$
 - $n/2==0 \ \&\& \ !(n/5==0)$
- Subprogramul f este definit alăturat. Indicați valoarea $f(102030)$.

```
int f (int x)
{ if(x>20) return 20+f(x/10);
  return 2020;
}
```

 - 1010
 - 2020
 - 2100
 - 3200
- Utilizând metoda backtracking, se generează toate numerele impare de cel mult trei cifre din mulțimea $\{0, 1, 2, 3\}$. Primele 8 soluții generate sunt, în această ordine: 1, 101, 103, 11, 111, 113, 121, 123. Cea de a 12-a soluție generată este:
 - 13
 - 31
 - 133
 - 201
- Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați” $(2, 8, 2, 9, 8, 9, 0, 7, 7, 9)$. Indicați câte dintre nodurile arborelui au exact doi fii.
 - 2
 - 3
 - 5
 - 6
- Un graf neorientat cu 20 de noduri are 100 de muchii. Numărul de muchii ce trebuie adăugate, pentru ca graful obținut să fie complet, este:
 - 10
 - 50
 - 90
 - 100

SUBIECTUL al II-lea (40 de puncte)

- 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 .

 - Scrieți valoarea care se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 12345, 780, 921, 4013, 75, 100214. (6p.)
 - Dacă pentru n se citește numărul 49, scrieți două seturi de date care pot fi citite în continuare astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 49. (6p.)
 - Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
 - Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să conțină o singură instrucțiune repetitivă. (6p.)

```
citește n (număr natural)
p←1; m←0; k←0
cât timp n≠0 execută
| citește x (număr natural)
| pentru i←1, k execută
| | x←[x/10]
| ■
| dacă x≠0 atunci c←x%10
| altfel c←n%10
| ■
| m←c*p+m
| n←[n/10]
| p←p*10; k←k+1
| ■
scrie m
```

2. Variabila t memorează coordonatele reale (abscisa și ordonata), în planul xOy , ale fiecăruia dintre cele trei vârfuri A , B și C ale unui triunghi. Știind că expresiile $C/C++$ de mai jos au ca valori abscisa vârfului A respectiv ordonatele vârfurilor B și C ale triunghiului, scrieți definiția unei structuri cu eticheta `triunghi`, care permite memorarea datelor precizate, și declarați corespunzător variabila t .
- `t.A.x` `t.B.y` `t.C.y` (6p.)

3. În secvența alăturată, variabila a memorează un șir cu cel mult 100 de caractere, iar variabilele i și k sunt de tip întreg. Scrieți ce se afișează pe ecran în urma executării secvenței.
- ```
k='a'-'A';
strcpy(a,"VICTorie");
cout<<strlen(a); | printf("%d", strlen(a));
for(i=0;i<strlen(a);i++)
 if(a[i]>='A' && a[i]<='Z') a[i]=a[i]+k;
 else a[i]=a[i]-k;
cout<<a; | printf("%s", a);
```
- (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul `putere` are trei parametri:
- $n$ , prin care primește un număr natural din intervalul  $[1, 10^9]$ ;
  - $d$  și  $p$ , prin care furnizează divizorul prim,  $d$ , care apare la cea mai mare putere,  $p$ , în descompunerea în factori primi a lui  $n$ ; dacă există mai mulți astfel de divizori se afișează cel mai mare dintre ei.
- Scrieți definiția completă a subprogramului.
- Exemplu:** dacă  $n=10780$ , atunci, în urma apelului,  $d=7$  și  $p=2$  ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). (10p.)
2. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ ,  $n$  și  $k$ , și construiește în memorie un tablou bidimensional cu  $n$  linii și  $n \cdot k$  coloane, numerotate începând cu 1, astfel încât fiecare linie  $i$  ( $i \in [1, n]$ ) memorează un șir crescător de termeni cu proprietatea că primul termen este  $i$ , fiecare valoare apare în șir de exact  $k$  ori și oricare doi termeni alăturați au valori egale sau consecutive.
- Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu valorile aflate pe aceeași linie separate prin câte un spațiu.
- |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 |
| 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 |
| 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 |
| 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 8 | 8 | 8 |
- Exemplu:** dacă  $n=4$  și  $k=3$ , se afișează pe ecran tabloul alăturat. (10p.)
3. Se consideră șirul  $1, 1, 2, 5, 13, 34, 89, 233, 610 \dots$  definit astfel:  $f_1=f_2=1$ ,  $f_n=3 \cdot f_{n-1}-f_{n-2}$  (unde  $n$  este un număr natural  $n \geq 3$ ):
- Se citesc de la tastatură două numere naturale  $x$  și  $y$  ( $x \leq y \leq 10^9$ ), valorile a doi termeni aflați pe poziții consecutive în șirul dat, și se cere să se scrie în fișierul text `bac.txt`, în ordine descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu  $y$ . Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
- Exemplu:** dacă se citesc numerele `89 233`
- fișierul `bac.txt` conține numerele `233 89 34 13 5 2 1 1`
- a. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)
- b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)