

Examenul de bacalaureat național 2019
Proba E. d)
Informatică
Limbajul C/C++

MODEL

Filieră teoretică, profil real, specializare științe ale naturii

- 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.

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 întregă n memorează un număr natural. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în n este divizibil cu 20, dar **NU** și cu 19.
 - $n \% 380 == 0 \ \&\& \ n / 20 == 0$
 - $n \% 380 != 0 \ || \ !(n \% 19 == 0)$
 - $n \% 20 == 0 \ \&\& \ n / 19 == 0$
 - $!(n \% 20 != 0 \ || \ n \% 19 == 0)$
- Interclasând descrescător tablourile unidimensionale (1, 5, 7, 10, 20) și (19, 10, 9, 8, 5, 1) se obține:
 - (20, 19, 10, 9, 8, 5, 1)
 - (20, 19, 10, 10, 9, 8, 7, 5, 5, 1, 1)
 - (20, 19, 10, 9, 8, 7, 5, 1)
 - (20, 19, 10, 10, 9, 7, 8, 5, 5, 1, 1)
- Indicați expresia care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă x aparține intervalului (-19, 19).
 - $\text{abs}(-x) < 19$
 - $\text{abs}(x) > -19$
 - $\text{abs}(-x) >= -19 \ \&\& \ \text{abs}(x) <= 19$
 - $\text{abs}(x) >= -19 \ || \ \text{abs}(-x) <= 19$
- Variabilele i și j sunt de tip întreg. Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze numerele de mai jos, în această ordine.

```
for (i=1; i<=4; i++)
{ for (j=1; j<=4; j++)
  cout<<.....<<" "; | printf("%d ", .....);
  cout<<endl; | printf("\n");
}
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

 - $i + (j - 1) * 4$
 - $i + (j + 1) * 4$
 - $(i - 1) * 4 + j$
 - $(i + 1) * 4 + j$
- Pentru orice valori naturale nenule ale variabilelor x și y , valoarea obținută pentru variabila z în urma executării secvenței este:

```
z<-x
cât timp z>=y execută z<-z-y
```

 - câtul împărțirii lui y la x
 - câtul împărțirii lui x la y
 - restul împărțirii lui y la x
 - restul împărțirii lui x la y

SUBIECTUL al II-lea (40 de puncte)

- Se consideră algoritmul alăturat, reprezentat în pseudocod.
 - Scrieți valorile afișate dacă se citește numărul 7. (6p.)
 - Scrieți cel mai mic și cel mai mare număr care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, ultima valoare afișată să fie 10. (6p.)
 - Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
 - Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind una dintre structurile `cât timp...execută` cu o structură repetitivă de alt tip. (6p.)
- ```
citește n
(număr natural)
k<-1
cât timp n>=1 execută
 dacă n>k atunci i<-k
 altfel i<-n
 n<-n-i
 cât timp i>=1 execută
 scrie k, ' '; i<-i-1
 k<-k+1
```

2. Variabilele **pre** și **pim** memorează partea reală, respectiv partea imaginară ale unui număr complex (numere reale). Declarați variabilele precizate și scrieți o secvență de instrucțiuni C/C++ care afișează pe ecran mesajul **multimea R**, dacă numărul complex corespunzător aparține mulțimii numerelor reale, sau mesajul **multimea C**, în caz contrar. (6p.)
3. Aplicând metoda căutării binare pentru a verifica dacă într-un tablou unidimensional există elementul cu o valoare oarecare **x**, aceasta se compară cu maximum 3 elemente ale tabloului. Dați exemplu de un astfel de tablou. (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Se citește un număr natural, **n**, și se cere să se scrie suma cifrelor prime ale lui **n**.  
**Exemplu:** dacă **n=1235405**, atunci se scrie **15**, iar dacă **n=140**, atunci se scrie **0**.  
Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. (10p.)
2. Scrieți un program C/C++ care citește de la tastatură numărul natural **n** ( $n \in [2, 10^2]$ ), apoi cele **n** elemente ale unui tablou unidimensional, numere naturale din intervalul  $[0, 10^2]$ , și afișează pe ecran, separate printr-un spațiu, primul număr impar și ultimul număr par memorate în tablou. Dacă nu există două astfel de numere, se afișează pe ecran mesajul **nu exista**.  
**Exemplu:** pentru **n=7** și tabloul **(8, 2, 0, 5, 9, 4, 1)** se afișează pe ecran numerele **5 4** iar pentru **n=7** și tabloul **(9, 3, 1, 5, 5, 9, 1)** se afișează pe ecran mesajul **nu exista** (10p.)
3. Un interval este numit **prieten de grad n** al unui șir dacă toate cele **n** numere întregi care aparțin intervalului sunt valori ale unor termeni ai șirului.  
Fișierul **bac.txt** conține un șir de cel mult  $10^6$  numere naturale din intervalul  $[0, 10^2]$ , separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul maxim **n** cu proprietatea că există un interval prieten de grad **n** al șirului aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.  
**Exemplu:** dacă fișierul conține numerele  
**10 10 11 3 4 2 49 4 2 3 21 2 27 11 10 14 15 5**  
atunci se afișează pe ecran **4** (toate cele **4** numere întregi din  $[2, 5]$  sunt termeni ai șirului).  
**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)