

Examenul de bacalaureat național 2020

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

Testul 5

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.

- Jelölje meg azt a C/C++ kifejezést, amelynek az értéke akkor és csak akkor 1, ha az x egész változóban tárolt természetes szám pontosan egy számjegyű.
 - $x/10==0$
 - $x\%10==0$
 - $(x/10)/10==0$
 - $(x\%10)\%10==0$
- Adott a mellékelt **f** alprogram. Mit fog kiírni a következő függvény meghívásra.
`f(3);`

```
void f(int n)
{
    int i;
    for(i=n; i>=1; i--)
    {
        f(n-1);
        cout<<i; | printf("%d", i);
    }
}
```

 - 121321
 - 1211312111212111
 - 322111
 - 3211122111112111
- Használva a backtracking módszerét generáljuk az összes lehetséges felöltözési sorrendeket. A ruhadarabokat következő halmazból vehetjük {ing, nyakkendő, nadrág, cipő, kabát, zokni}. A generálásnál figyelembe kell venni a következő megkötéseket: az inget a nyakkendő előtt kell felvenni, a nyakkendőt a kabát előtt kell felvenni és a zoknit a nadrág és a cipő előtt kell felvenni. Az első három generált megoldás a következő: (ing, nyakkendő, nadrág, kabát, zokni, cipő), (ing, nyakkendő, nadrág, zokni, cipő, kabát), (ing, nyakkendő, nadrág, zokni, kabát, cipő). Jelöljük meg a hatodik generált megoldást.
 - (ing, nyakkendő, kabát, zokni, nadrág, cipő)
 - (ing, nyakkendő, zokni, nadrág, kabát, cipő)
 - (ing, nyakkendő, zokni, nadrág, cipő, kabát)
 - (ing, nyakkendő, zokni, kabát, nadrág, cipő)
- Egy 9 csúcsponttal rendelkező fa, csúcspontjai 1-től 9-ig vannak megszámozva és a következő "apák" tömbjével van megadva: (2, 7, 0, 8, 1, 5, 3, 9, 2). Melyik csúcspont a fa gyökere?
 - 1
 - 3
 - 4
 - 6
- Egy 2020 csúcsponttal rendelkező nemirányított gráf szomszédossági mátrixa 200 nem nulla elemet tartalmaz. A gráf összefüggő komponenseinek maximális száma:
 - 2006
 - 2000
 - 1820
 - 400

SUBIECTUL al II-lea

(40 de puncte)

1. A mellékelt algoritmus pszeudokódban van megadva.

Az $a \% b$ az a természetes szám és a b nem nulla természetes szám osztási maradékát, a $[c]$ a valós c szám egész részét jelöli.

a. Mit fog kiírni az algoritmus, ha a beolvasott szám 100? **(6p.)**

b. Írjuk le az összes olyan számot az $[1, 9]$ intervallumból amelyeket, ha beolvassunk akkor az algoritmus N -t fog kiírni. **(6p.)**

c. Írjuk meg a mellékelt algoritmust a C/C++-ban. **(10p.)**

d. Írjuk meg pszeudokódban a mellékelt algoritmussal egyenértékű algoritmust úgy, hogy ne használjunk semmilyen ismétlődő struktúrát. **(6p.)**

```
citește n
    (număr natural nenul)
x ← 1; y ← n; d ← 2
cât timp x < y execută
    dacă n % d = 0 atunci
        x ← d
        y ← [n/d]
    d ← d + 1
dacă x = y atunci
    scrie 'D', x
altfel scrie 'N'
```

2. Az s változó egyszerre tárolja egy complex egyenlet megoldásainak a számát (természetes szám $[2, 10^2]$) és a megoldásait (valós részt és imaginárius részt, mindkettő valós szám). Tudva azt, hogy az alábbi kifejezések a megoldások számát illetve az első megoldás valós és imaginárius részét tárolja, írjunk fel egy struktúrát **ecuatie** néven, amelyik képes tárolni a fent említett adatokat egy s nevű változóban.

`s.numar` `s.solutie[0].pre` `s.solutie[0].pim`

(6p.)

3. Az i és a j változók egész típusú változók, az a változó egy mátrix 6 sorral és 6 oszloppal, 0-tól 5-ig sorszámozva, kezdetben mindegyik mátrix elem `@` karaktert tartalmaz. Más változók használata nélkül írjunk a pontok helyére egy utasítássorozatot, amely a végrehajtás után a mintának megfelelően feltölti az a mátrixot.

```
for (i=0; i<6; i++)
    for (j=0; j<6; j++)
        .....
```

```
( ( ( ) ) )
* ( ( ) ) *
* * ( ) * *
* * ( ) * *
* ( ( ) ) *
```

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Egy szám b ($b \leq 10$) számrendszerbe van írva, ha minden számjegye a $[0, b-1]$ intervallumból van. A **baza** alprogramnak egy n paramétere van, amelyen keresztül kap egy n természetes számot. ($n \in [0, 10^9]$). Az alprogram visszatéríti a legkisebb számrendszert a $[2, 10]$ intervallumból amelyikbe létezik az n szám. Írjuk meg a teljes alprogramot.

Példa: ha $n=50731$, az alprogram **8** térít vissza.

(10p.)

2. Egy maximum 100 karaktert tartalmazó szöveg szavakat és számokat tartalmaz egy-egy szóközzel elválasztva. A szavak csak az angol abc kisbetűit tartalmazza, a számok valósak és pozitívak. A számokban az egész és a tizedes rész vesszővel van elválasztva vagy csak egész része van a számnak, mint a példában. Írjunk egy C/C++ programot ami beolvas a billentyűzetről egy szöveget és kiírja, hogy hány egész szám van benne.

Példa: a következő szövegre

`grus leucogeranus are 1,40 m inaltime si traieste intre 30 si 40 de ani`
kiírja, hogy: 2

(10p.)

3. A **bac.txt** fájl maximum 10^6 darab egész számot tartalmaz a $[-10^3, 10^3]$ intervallumból, egy-egy szóközzel elválasztva. Írjuk ki a képernyőre az egymást követő számok összegének a maximumát a fájlból. Tervezzük meg az algoritmust, úgy hogy memória felhasználás és futási idő szempontjából is hatékony legyen.

Példa: ha a **bac.txt** a következő számokat tartalmazza `4 -6 7 2 -1 4 -10 -3 9 2 -2`
A program a képernyőre kiírja, hogy: 12

a. Írjuk meg a megtervezett algoritmust C/C++-ban.

(8p.)

b. Írjuk le szavakba az algoritmusunkat és írjuk le, hogy miért hatékony.

(2p.)