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

Testul3

Filieră teoretică, profil real, specializare matematică-informatică /matematică-informatică intensiv informatică
Filieră vocaţională, profil militar, specializare matematică-informatică

- **Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.**
- **Timpul de lucru efectiv este de 3 ore.**
- **Identificatorii utilizaţiîn rezolvări trebuie să respecte precizările din enunţ (bold), iar în lipsa unor precizări explicite, notaţiile trebuie să corespundă cu semnificaţiile 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.**

I. TÉTEL **(20 pont)**

Írja a vizsgalapra az 1 és 5 közti feladatokra adott helyes válaszoknak megfelelő betűt. Mindenik helyes válasz 4 pontot ér.

1. Az x és y egész változók mindegyike nullától különböző természetes számot tartalmaz. Határozzátok meg azt a C/C++ kifejezést melynek az értéke 1 akkor és csakis akkor, ha az x változóban tárolt érték egyeseinek számjegye 2, és x többszöröse az y változóban tárolt értéknek?

a. `x%2==10 || y%x==0` **b.** `x%10==2 || y%x==0`

c. `x%2==10 && x%y==0` **d.** `x%10==2 && x%y==0`

2. Az f alprogram mellékelt definíciója esetén, mi lesz kiírva az `f(3)` hívás következtében?

```
void f(int x)
{cout<<x;  |  printf("%d",x);
while(x>0){f(x-1); x=x-1;}
}
```

a. `321021010` **b.** `32100100` **c.** `3210` **d.** `321`

3. A backtracking módszert használva generáljuk az összes lehetséges megoldást, hogy 4 értékes kőből álló fűzéreket készítsünk. A köveket a `{rubin,opal,safir,smarald,topaz}` halmazból választhatjuk ki, úgy hogy két egymás melletti pozícióban ne szerepeljenek a `{rubin,safir,topaz}` részhalmazból kiválasztott kövek. Az első 8 megoldás, a generálás sorrendjében: `(rubin,opal,rubin,opal)`, `(rubin,opal,rubin,smarald)`, `(rubin,opal,opal,rubin)`, `(rubin,opal,opal,opal)`, `(rubin,opal,opal,safir)`, `(rubin,opal,opal,smarald)`, `(rubin,opal,opal,topaz)`, `(rubin,opal,safir,opal)`.
Az utolsó generált fűzér:

a. `(topaz,smarald,topaz,topaz)` **b.** `(topaz,smarald,topaz,opal)`

c. `(topaz,smarald,topaz,smarald)` **d.** `(topaz,smarald,smarald,topaz)`

4. Adott egy 10 csomópontot tartalmazó gyökeres fa, melyben a csomópontokat 1-től 10-ig sorszámozzuk. A fa ősvektora, az "apák" tömbje: `t=(2,5,1,1,0,3,3,7,4,6)`. Adjátok meg a fa leveleinek számát.

a. `3` **b.** `4` **c.** `5` **d.** `6`

5. Adott egy 10 csomópontot tartalmazó irányítás nélküli gráf, csomópontjait 1-től 10-ig sorszámozzuk, élei: `[1,2]`, `[2,3]`, `[2,10]`, `[3,10]`, `[4,5]`, `[4,6]`, `[5,6]`, `[6,9]`, `[7,8]`, `[7,9]`, `[8,9]`.
Határozzátok meg legkevesebb hány élt kell hozzáadni ahhoz, hogy a létrejött gráf euler féle legyen.

a. `1` **b.** `2` **c.** `3` **d.** `4`

II. TÉTEL (40 pont)

1. **Adott a mellékelt algoritmus pszeudokódban.**
Az **a%b** jelöli az a természetes szám b nem nulla természetes számmal való osztási maradékát, és **[c]** a **c** valós szám egész részét.

a. Írjátok le mit ír ki, ha a beolvasott szám **2754578**. **(6p.)**

b. Adjatok meg két olyan számot, amelyeket beolvashatunk, és ha elvégezzük az algoritmust, az első szám esetén az **1 0** lesz kiírva, a második esetén a **-1 0**. **(6p.)**

c. Írjatok az adott algoritmusnak megfelelő C/C++ programot. **(10p.)**

d. Írjátok meg azt az eredetivel egyenértékű algoritmust pszeudokódban, amelyikben az **ismételd...ameddig** struktúrát helyettesítitek egy másik ismétlő struktúrával. **(6p.)**

```
Beolvas (n>9 természetes szám)
 c1←n%10; n←[n/10]; c2←n%10
┌ha c1=c2 akkor s←0
│ │különben
│ ┌ha c1>c2 akkor s←1
│ │ különben s←-1
│ └■
└■
┌ismételd
│ c1←n%10; n←[n/10]; c2←n%10
└└ameddig (c1-c2)*s≤0
kiír s,'',n
```

2. A **fig** változó egy kör adatait tárolja: az **xOy** koordináta rendszerben a kör középpontjának valós koordinátáit (abszcissza és ordináta), illetve a kör sugarának hosszát. Tudva, hogy az alábbi C/C++ kifejezések értékei olyan valós számok, amelyek a kör adatait tárolják. Írjátok meg a **cerc** címkéjű struktúra definícióját, úgyhogy, abban tárolhassuk a rögzített adatokat, és deklaráljátok helyesen a **fig** változót.
fig.centru.x fig.centru.y fig.raza **(6p.)**

3. A **p** változó egész típusú, az **s** változó egy olyan karakterlánc, amely legtöbb 20 karaktert, csak az angol ábécé nagybetűit tárolhatja. Más változó használata nélkül, írjunk olyan kifejezéseket, amelyek hatására a képernyőn megjelenik az **s**-ben tárolt karakterlánc összes betűje a **A** magánhangzó kivételéve, abban az esetben ha az egy **I** magánhangzó mellett található az eredeti karakterláncban. A betűk a karakterláncban való előfordulás sorrendjében legyenek kiírva.
Például: ha az **s**-ben tárolt karakterlánc **ALIANTA** vagy **ALAINTA** a képernyőn **ALINTA** jelenik meg. **(6p.)**

III. TÉTEL (30 pont)

1. A **factori** alprogramnak két paramétere van: **n** és **m**, melyeken keresztül egy-egy természetes számot kap a **[1,10⁹]** intervallumból. Az alprogram visszatéríti azon prím értékek számát, amelyeket az **n** szám prímtényezőre bontásában is és ugyanakkor az **m** szám prímtényezőre bontásában is megtalálhatók.
Írjátok mega z alprogram teljes definícióját.
Például: ha **n=750** és **m=490**, akkor az alprogram a **2-t** téríti vissza (**750=2·3·5³**, **490=2·5·7²**). **(10p.)**

2. Írjatok egy **C/C++** programot, amelyik a billentyűzetről beolvas egy **n**(**n∈[2,20]**) természetes számot, és felépíti a memóriában az **n** sorból és **n** oszlopból álló kétdimenziós tömböt, úgyhogy:
- A mellékátlón található összes elem értéke nulla;
- Mindegyik sorban a mellékátlótól kezdődően jobbról balra egymásutáni, szigorúan növekvő értékek legyenek, és a mellékátlótól kezdődően balról jobbra szintén egymásutáni szigorúan növekvő értékek legyenek.

A program az így kapott tömböt kiírja a képernyőre minden sorát a képernyő külön sorába, a sorokban minden elemet egy-egy szóközzel elválasztva.
Például: ha **n=5** a képernyőn a mellékelt tömb jelenik meg **(10p.)**

```
4 3 2 1 0
3 2 1 0 1
2 1 0 1 2
1 0 1 2 3
0 1 2 3 4
```

3. A **bac.in** állomány egy legtöbb **10⁶** egész számokból álló számsort tartalmaz. A számok a **[-10⁹,10⁹]** intervallumból vannak, egy-egy szóközzel vannak elválasztva és található köztük legalább egy negatív szám.
Írjátok ki a képernyőre egy olyan maximális hosszúságú részsornak a hosszát, amelyik vagy egy negatív számmal kezdődik, vagy egy negatív számmal végződik. Egy részsorozat a számsor egymásután elhelyezkedő tagjaiból áll, a hossza pedig megegyezik ezen tagok számával.
Tervezzetek a futási idő és a memóriahasználat szempontjából hatékony algoritmust.
Például: ha az állományban található számok:**12 25 -6 7 80-75 101 -652-12487 99 210**
A képernyőn a **11** jelenik meg (**-6 7 80 -75 101 -6 52 -124 87 99 210** részsorozatnak megfelelően).
a. Írjatok a tervezett algoritmusnak megfelelő C/C++ programot. **(8p.)**

b. Írjátok le saját szavaitokkal a használt algoritmust, és indokoljátok annak hatékonyságát. **(2p.)**