

Testul 4

Filiera teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filiera vocatională, 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 muchie are extremități distincte și oricare două muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Az 1-től 5-ig számozott itemek esetén írja a vizsgalapra a helyes válasznak megfelelő betűt! Minden helyes válasz 4 pontot ér.

- Az **x** és **y** egész típusú változók, az **x** 8-as az **y** pedig a 6-os értéket tárolja. Jelölje be azt a C/C++ kifejezést amelyik értéke 0!
 - $3 * x - 4 * y == 0$
 - $(x + y) / 2 > x * y + 1$
 - $!(x / 2 + 2 == y)$
 - $x - y + 3 != 0$
- Tekintsük a mellékelt **f** alprogramot.
`int f(int n)`
Jelölje be az `f(38627)` visszatérési értékét!
`{ int c;`
`if (n==0) return 9;`
`c=f(n/10); if (n%10<c) return n%10;`
`return c;`
`}`
 - 2
 - 3
 - 7
 - 8
- A backtracking módszert használva generáljuk növekvő sorrendbe az összes 5 számjegyű számot az **{1,2}** halmaz elemeiből, azzal a megkötéssel, hogy egymás után nem szerepelhet több mint két darab 1-es számjegy.
Az első 5 generált megoldást a következő sorrendbe kapjuk: 11211, 11212, 11221, 11222, 12112.
Jelölje be a 8. generált megoldást!
 - 12122
 - 12211
 - 12212
 - 12221
- Egy fának 10 csomópontja van 1-től 10-ig való sorszámozással, az élei a következők: [1,2], [1,3], [1,5], [1,6], [2,8], [2,9], [3,4], [3,10], [4,7]. Jelölje be annak az elemi útnak a hosszát, amelynek a végpontjai a 6-os és 7-es csúcspontok!
 - 1
 - 2
 - 3
 - 4
- Jelölje be az 5 csúcspontból álló különböző nemirányított gráfok számát! Két gráf különbözik, ha a szomszédossági mátrixaik különbözöek.
 - 5^4
 - 5^2
 - 2^{10}
 - 4^{10}

SUBIECTUL al II-lea (40 de puncte)

1. A mellékelt algoritmus pszeudokódban van ábrázolva.
 Az $a \% b$ az a természetes szám és a b nem nulla természetes szám osztási maradékát jelöli, a $[c]$ a c valós szám egész részét jelöli.
 - a. Az algoritmus, a végrehajtása során mit fog kiírni, ha a beolvasott értékek: 296385 és 3. (6p.)
 - b. A k beolvasott értéke 4, adja meg az n legkisebb és a legnagyobb értéket a $[10^4, 10^5)$ intervallumból, úgy, hogy az algoritmus végrehajtása során a kiírt szám 1 legyen. (6p.)
 - c. Írja meg az algoritmusnak megfelelő C/C++ programot! (10p.)
 - d. Írja meg az adott algoritmusnak megfelelő pszeudokódot úgy, hogy az **amíg ... végezd el** struktúrát egy utótesztelésű ciklusra cseréli! (6p.)

2. A **c** változó adatokat tárol egy csokoládéről, a súlyát és a garancia dátumát. Tudva azt, hogy az alábbi C/C++ kifejezések természetes számok és a súlyt és a dátumot (nap, hónap és év) jelölik, írja le egy olyan strukturának a definícióját ciocolata címkével, amelyik a fent említett adatokat tárolja a **c** változóban!
c.grama_j c.datae.zi c.datae.luna c.datae.an (6p.)
3. A mellékelt szekvenciában az **i** változó egész típusú, az **s** és a **t** változók pedig egy-egy 20 karakterből álló karakterláncok. Mit fog kiírni a szekvencia a végrehajtása során? (6p.)
- ```
strcpy(s, "sanataTEA");
cout<<strlen(s1)<<endl; |
printf("%d\n", strlen(s1));
i=0;
while(i<strlen(s))
 if(s[i]=='a')
 { strcpy(t, s+i+1); strcpy(s+i, t); }
 else i=i+1;
cout<<s; | printf("%s", s);
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Egy online játékot **n** zsetonnal és **k** ( $k \geq 2$ ) játékoskal lehet játszani, ha az **n** darab zsetont egyenlően el lehet osztani a **k** játékos között.  
A joc alprogramnak egy **n** paramétere van ( $n \in [2, 10^4]$ ) ami a zsetonok számát jelöli. Az alprogram visszatéríti a **k** lehetséges értékeinek a számát. Írja meg a teljes alprogramot!  
**Példa:** ha **n=12**, akkor az alprogram **5-öt** térít vissza (mert a 12 zsetont egyenlően szét lehet osztani 2,3,4,6 és 12 játékos között). (10p.)
2. Írjon egy C/C++ programot, amely a billentyűzetről természetes számokat olvas be a  $[2, 10^2]$ : intervallumból, egy **m**, **n** és egy kétdimenziós **m** soros és **n** oszlopos tömb elemeit! A program minden sorból kiírja a legnagyobb elemet, amely kisebb, mint 21 vagy a nu exista üzenetet, ha ilyen nincs. A számokat és az üzeneteket soronként kell kiírni a táblázat sorainak a sorrendjében.

**Példa:** ha **m=3**, **n=5** és a mellékelt táblázat esetén a következőt írja ki a program

|           |    |    |    |    |    |
|-----------|----|----|----|----|----|
| 16        | 6  | 16 | 21 | 4  | 90 |
| nu exista | 92 | 26 | 36 | 95 | 80 |
| 9         | 5  | 2  | 9  | 7  | 3  |

(10p.)

3. A bac1.txt és bac2.txt természetes számokat tartalmaz a  $[1, 10^5]$ : intervallumból. A bac1.txt első sora egy **n1** számot tartalmaz, a második sora pedig **n1** darab számot tartalmaz, a bac2.txt fájl első sora egy **n2** számot tartalmaz a második sora pedig **n2** darab számot tartalmaz. A fájlokban minden egyes sor elemei növekvő sorrendbe vannak elrendezve, és egy szóközzel vannak elválasztva egymástól. Írja ki a képernyőre egy-egy szóközzel elválasztva azokat a számokat, amelyek **5**-el oszthatóak, és csak az egyik fájlban szerepelnek! Ha nem létezik ilyen szám, akkor írja ki, hogy: nu exista! Tervezzon meg egy olyan algoritmust, amely memória és végrehajtási idő szempontjából hatékony!

**Példa:** ha a **bac1.txt** a

|                 |                        |                       |
|-----------------|------------------------|-----------------------|
| 7               | ha a <b>bac2.txt</b> a | 9                     |
| 1 2 3 4 7 20 60 | következő számokat     | 3 5 7 8 9 10 12 20 24 |
| tartalmazza     | tartalmazza            |                       |

A képernyőn megjelennek ezeket a számok a következő sorrendben: 5 10 60.

- a. Írja le magyarul a megtervezett algoritmust és indokolja meg annak hatékonyságát! (2p.)  
b. Írja meg C/C++ nyelven a megtervezett algoritmust! (8p.)