

**III. tétel (30 pont)**

**Az 1-es feladat esetén, a helyes válasznak megfelelő betűt, írástok a vizsgalagra.**

1. Egy program beolvas, egy zérótól különböző természetes számot az  $n$ -nek, majd legenerálja és lexicografikailag csökkenő sorrendbe, megjeleníti az összes  $n$  elemű kombinációját a  $\{0,1\}$  halmaznak. Így,  $n=2$ -re a kombinációk a következő sorrendbe jelennek meg: 11, 10, 01, 00. Ha futtatjuk ezt a programot és az  $n$ -nek értéként a 8 -ast adjuk, akkor a 10101000 kombináció után rögtön az: (4p.)
- a. 01010111                      b. 10100111                      c. 10101001                      d. 10100100

**Az alábbi feladatok esetén a kérésekre, kérdésekre a választ írástok a vizsgalagra!**

2. Mellékelve van, az `f` függvény definíciója. Írástok 4 értéket, melyet felvehet az  $n$  úgy, hogy, ha 4-szer meghívjuk ezekkel az értékekkel az `f` függvényt, akkor páronként 4 különböző értéket kapunk. (6p.)
- ```
int f(int n)
{
    if (n<=9) return 0;
    if (n%4==0) return 0;
    return 1+f(n-3);
}
```

```
int f(int n)
{
    if (n<=9) return 0;
    if (n%4==0) return 0;
    return 1+f(n-3);
}
```
3. A `verif` függvény az `a`, `b` és `c` paramétereken keresztül, kap, három zérótól különböző természetes számot, mindegyik szám legtöbb négy számjegyű. A függvény visszatéríti az 1 -est, ha e három érték, egy háromszög oldalhosszainak felelnek meg, ellenkező esetben a 0 -ást téríti vissza.
- a) Írástok le, a teljes definícióját a `verif` függvénynek.
- b) Írástok `C/C++` programot, mely a billentyűzetről beolvas, hat zérótól különböző, maximum 4 számjegyből álló természetes számot, majd a `verif` függvény meghívásával ellenőrzi, hogy az első három beolvasott érték, egy háromszög oldalhosszai e vagy sem, és ugyanezt ellenőrzi az utolsó három beolvasott értékre is; ha mindkét esetben igaz az állítás és a két háromszög kongruens, akkor kiírja a képernyőre a `congruente` üzenetet, vagy a `necongruente` üzenetet, ha a két háromszög nem kongruens; de ha a csoportosításból valamelyik triplet nem alkot háromszöget, akkor a `nu` üzenetet jeleníti meg. (10p.)
4. A `BAC.DAT` állomány első sorába, szóközzel elválasztva, van két természetes szám, az  $n$  és az  $m$  ( $2 \leq n \leq 1000$ ,  $2 \leq m \leq 1000$ ) értéke, a második sorába van  $n$  darab egész érték és a harmadik sorába van  $m$  darab egész érték. Az állomány második és harmadik sorába az értékek szigorúan növekvő sorrendbe vannak beírva, mindegyik érték legtöbb 4 jegyű és egymástól szóközzel vannak elválasztva.
- Az állomány harmadik sorából azt a két egymás utáni értéket kell kiíratni a képernyőre, amelyek által meghatározott zárt intervallum tartalmazza a legtöbbet az állomány második sorában lévő értékekből. Használjatok egy, a memória-helyfoglalás és a végrehajtási idő szempontjából hatékony módszert.
- Példa:** ha a `BAC.DAT` állomány a mellékel értékeket tartalmazza, akkor a program megjeleníti a:
- ```
10 4
-1 1 3 4 5 6 10 15 16 117
0 1 9 20
```

```
10 4
-1 1 3 4 5 6 10 15 16 117
0 1 9 20
```
- Magyarázat: az állomány harmadik sorába lévő négy érték három intervallumot határoz meg:  $[0,1]$ ,  $[1,9]$ ,  $[9,20]$ ; az  $[1,9]$  intervallumban található 5 érték az állomány második sorába lévő értékek közül, és a három intervallumban talált értékek száma közül, ez az 5 -ös a legnagyobb érték.
- a) Írástok le röviden, természetes nyelvvel, a használt módszert, megmagyarázva ennek a hatékonyságát (3 – 4 sor) (4p.)
- b) Írástok `C/C++` programot, mely a leírt módszerrel megoldja a feladatot. (6p.)