

III. Tétel (30 pont)

Az első pontnál írjátok a vizsgalapra a helyes válasznak megfelelő betűt.

1. A backtracking eljárást felhasználva generáljuk az összes **negyedrendű** négyzetes mátrixot, melynek elemeit a $\{0,1\}$ halmazból vesszük, azzal a tulajdonsággal, hogy az 1-es érték minden sorban és oszlopban csak **egyszer** forduljon elő. Az első 4 megoldás, a következő sorrendben generálható:

1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0
0 1 0 0	0 1 0 0	0 0 1 0	0 0 1 0
0 0 1 0	0 0 0 1	0 1 0 0	0 0 0 1
0 0 0 1	0 0 1 0	0 0 0 1	0 1 0 0

Melyik lesz a **nyolcadik** megoldás?

(4p.)

a. 0 1 0 0	b. 0 1 0 0	c. 0 1 0 0	d. 0 0 1 0
1 0 0 0	1 0 0 0	0 0 1 0	1 0 0 0
0 0 0 1	0 0 1 0	1 0 0 0	0 1 0 0
0 0 1 0	0 0 0 1	0 0 0 1	0 0 0 1

Írjátok a vizsgalapra a válaszokat az alábbi kérdések mindegyikére.

2. Tekintsük a mellékelve definiált `int f(unsigned int n)` alprogramot.
- ```
{ if (n>20) return 0;
 else return 5+f(n+5); }
```
- a)** Mennyi az `f(25)` értéke?
- b)** És az `f(1)+f(5)+f(15)` értéke?
- (6p.)**
3. Tekintsük a **számok** alprogramot, amely az első **a** paraméter segítségével megkap egy maximum 8 számjegyű, nullától különböző természetes értéket és visszatéríti a második, **b** paraméter segítségével azt a legkisebb természetes számot, amelyet az **a** szám különböző számjegyeiből alkotni lehet.

**a)** Írjátok le a **számok** alprogram teljes definícióját.

**(4p.)**

**b)** Tekintsük az **adatok.be** szöveges állományt, melynek első sorában egy **n** ( $n \leq 100$ ) nullától különböző természetes szám található, a másodikban pedig **n** darab, nullától különböző, maximum 8 számjegyű természetes szám, egymástól egy-egy szóközzel elválasztva. Írjátok egy C/C++ programot mely kiolvassa az **adatok.be** szöveges állományból az összes számot és kiírja a képernyőre, egy-egy szóközzel elválasztva, a második sorból azokat a számokat, melyek számjegyei növekvő sorrendben vannak, felhasználva a **számok** alprogram meghívásait. Ha nem létezik egyetlen ilyen szám sem, a képernyőre a 0 érték íródjon ki.

**Például:** ha az **adatok.be** állomány a mellékelt adatokat tartalmazza, a képernyőre a következő számok íródhatnak ki:

**(6p.)**

|                           |
|---------------------------|
| 6                         |
| 16 175 333 242477 321 269 |

4. Írjátok egy C/C++ programot mely beolvassza a billentyűzetről egy nullától különböző természetes **n** ( $n \leq 100$ ) számot, és  $2 \cdot n$  természetes, maximum 3 számjegyű számot, melyek közül az első **n** az **a** elemeit, és a következő **n** a **b** vektor elemit képezi, majd a memóriában felépíti és kiírja a képernyőre egy-egy szóközzel elválasztva egy **c** vektornak az **n** elemét. A tömbök elemeinek indexelése 1-től kezdődik. A `c[i]` ( $1 \leq i \leq n$ ) elem értékét a következő képlet adja meg:

`c[i] = a[i]` összefűzve `b[i]-vel`, ha `a[i] < b[i]`  
          `b[i]` összefűzve `a[i]-vel`, ellenkező esetben

**Példa:** ha a beolvasott értékek  $n=3$ ,  $a=(12,123,345)$  és  $b=(1,234,15)$ , akkor a **c** tömb elemei lesznek kiírva a következők szerint:

112 123234 15345 **(10p.)**