

III. tétel (30 pont)

Az 1-es alponthoz írjátok a vizsgalapra a helyes válasznak megfelelő betűt.

1. Egy program felépíti és kiírja az $A \times B \times C$ Descartes szorzat elemeit, ahol $A=\{1,2,3,4\}$, $B=\{1,2,3\}$, $C=\{1,2\}$. A következő elemek közül melyiket **NEM** írja ki? (4p.)
a. (3,2,1) b. (1,3,2) c. (1,2,3) d. (2,2,2)

Írjátok a vizsgalapra a következő követelményekre a választ.

2. Ha a `sub` alprogram a mellékelt módon van meghatározva, állapítsátok meg mennyi lesz a `sub(4)` értéke. Hát a `sub(123986)` értéke? (6p.)
- ```
int sub(long n)
{
 if (n!=0)
 if (n%2!=0) return 1+sub(n/10);
 else return sub(n/10);
 else return 0;
}
```
3. a) Írjátok le csak a fejlécét a `prim` alprogramnak, amely az `n` paraméteren keresztül kap egy legfeljebb négy számjegyű természetes számot és visszatéríti a `p` paraméteren keresztül a `true` értéket, ha `n` prímszám és máskülönben `false` értéket. (2p.)  
b) Írjátok egy `C/C++` programot, amely beolvas a billentyűzetről egy `n` ( $3 < n < 10000$ ) természetes számot és kiírja a képernyőre szóközzel elválasztva az első `n` prímszámot, a `prim` alprogram szükséges meghívásait használva. (8p.)

**Például:** ha `n=4`, a képernyőre ki lesznek írva a 2 3 5 7 számok

4. A `bac.in` szöveges állomány első sorában van egy `n` ( $0 < n < 10000$ ) természetes szám és a következő sorban `n` darab természetes szám az  $[1,100]$  intervallumból. Olvassátok be az állományból az összes számot, írjátok ki a képernyőre növekvő sorrendben a számokat az állomány második sorából és mindegyiknek az előfordulási számát. Ha egy szám többször jelenik meg, csak egyszer lesz kiírva. Minden „érték – előfordulási szám” számpár a képernyő külön soraira lesz kiírva szóközzel elválasztva úgy, mint a következő példában. Válasszatok a futási idő szempontjából egy hatékony algoritmust a feladat megoldásához.

**Példa:** ha a `bac.in` állomány tartalma:

12  
1 2 2 3 2 2 3 3 2 3 2 1

A képernyőre ebben a sorrendben a következő számpárok lesznek kiírva:

3 4  
2 6  
1 2

- a) Természetes nyelvezetet használva írjátok le a használt megoldási módszert és magyarázzátok el, hogy miben áll a hatékonysága (4 – 6 sor). (4p.)  
b) Írjátok le az a) pontnál leírt módszert használva azt a `C/C++` programot, amely megoldja a feladatot! (6p.)