

**III. Tétel (30 pont)**

**Az 1-es pontban írástok a vizsgalapra a helyes válasz betűjelét.**

1. A backtracking módszert alkalmazva, képezzük az  $\{1, 2, 3, 4\}$  halmaz összes permutációját. Az így képzett első három permutáció rendre a következő: 1234, 1243, 1324. Melyik következik rögtön a 3412 után. **(4p.)**
- a. 3214                      b. 3413                      c. 4123                      d. 3421

**Az alábbi feladatok megoldásait írástok a vizsgalapra.**

2. Tekintsük a mellékelt `f` alprogramot. Mennyi az `f(7)` értéke? Hát az `f(100)`-é? **(6p.)**
- ```
function f(x:integer):integer;  
begin  
  if x mod 6=0 then f:=x  
    else f:=f(x-1)  
  end;
```
3. Írja meg a `P`, háromparaméteres alprogram teljes definícióját, amely az első `a` paraméterben megkap egy maximum 100 elemű egydimenziós tömböt, amelyben az elemek legfeljebb 4 számjegyű egész számok, a második `n` paraméterben megkapja a tömb elemeinek tulajdonképpen számát, valamint a `k` paraméterben egy természetes számot ( $k < 101$ ), és visszatéríti azt a legnagyobb összeget, amelyet a tömb `k` elemének összegzésével lehet kapni.
- Példa:** ha  $n=6$  és  $k=4$ , valamint a sorozat elemei (5, 2, 5, 4, 1, 3), akkor a hívás után visszatérített érték 17. **(10p.)**
4. A `numere.txt` szöveges állomány első sorában egy `n`, ( $0 < n < 100000$ ) természetes szám található, a következő sorban pedig, egy-egy szóközzel elválasztva, `n` darab legtöbb kétjegyű természetes szám.
- a) Írástok a futási idő szempontjából egy hatékony **Pascal** programot, amely eldönti, hogy az állomány második sorában levő számok szigorúan növekvő sorrendben vannak-e. Ha igen, akkor a **DA**, ellenkező esetben pedig a **NU** üzenetet írja ki a képernyőre.
- Példa:** Ha a `numere.txt` állomány tartalma:
- ```
7  
3 5 2 1 5 23 1
```
- Akkor a kiírt üzenet a **NU**. **(6p.)**
- b) Írástok le a saját szavaitokkal tömören az általatok használt megoldási módszert és magyarázzátok meg, hogy miben rejlik a hatékonysága (3 – 4 sorban). **(4p.)**