

### **III. Tétel (30 pont)**

**Az 1-es feladat esetén a helyes válasz betűjelét írja a vizsgalapra.**

1. A backtracking módszert használjuk arra, hogy az összes lehetséges módon felírjuk a 6-os számot legalább két, nullától különböző természetes szám összegeként. A összegben szereplő értékek növekvő sorrendben vannak. A generált megoldások sorrendben: 1+1+1+1+1+1, 1+1+1+1+2, 1+1+1+3, 1+1+4, 1+5, 2+2+2, 2+4 és 3+3. Ugyanezt a módszert használjuk a 9-es szám felírására. Hány darab 2+... alakú megoldás lesz generálva? **(4p.)**
- a. 2                      b. 3                      c. 4                      d. 5

**Az alábbi feladatok esetén a választ írja a vizsgalapra.**

2. A mellékelt programrészletben az `i`, `j` és `aux` változók egész típusúak, valamint `v` egy egydimenziós tömböt tárol, amely elemeinek értéke: `v[1]=2`, `v[2]=13`, `v[3]=1`, `v[4]=5`, `v[5]=-4`. A mellékelt programrész végrehajtása során hány csere történik? Adjon példát más 5 darab értékre úgy, hogy a mellékelt programrész végrehajtása során egyetlen csere se történjen. **(6p.)**
- ```
for(i=1;i<5;i++)
for(j=5;j>i;j--)
if(v[i]<v[j])
{ aux=v[i];
  v[i]=v[j];
  v[j]=aux;
}
```
3. a) A `bac.in` állomány első sorában egy `n` ( $n \leq 1000$ ), nullától különböző természetes szám van, a második sorában pedig `n` darab, legfeljebb 4 számjegyből álló természetes szám, egy-egy szóközzel elválasztva. Írjon egy C/C++ programot, amely az állományból beolvassa az értékeket, és a képernyőre kiírja a **DA** üzenetet, ha az állomány második sorában levő páros számok növekvő sorrendben, a páratlan számok pedig csökkenő sorrendben vannak, ellenkező esetben a **NU** üzenetet írja ki. Válasszon a memória-felhasználás szempontjából hatékony algoritmust. **(6p.)**
- b) Írja le röviden a saját szavaival az alkalmazott módszert, indokolva a hatékonyságát. **(4p.)**

**Példa:** a `bac.in` állomány mellékelt tartalma esetén, a képernyőre kiírt üzenet: **DA**

|                                                                                       |
|---------------------------------------------------------------------------------------|
| 8                                                                                     |
| <u>10</u> <u>1133</u> <u>12</u> <u>331</u> <u>42</u> <u>1354</u> <u>221</u> <u>13</u> |

4. a) Írja le az `sdiv` alprogram teljes definícióját, amely az `y` paraméterben megkap egy legfeljebb 6 számjegyű természetes számot, és visszatéríti az `y` összes osztójának összegét. **(5p.)**
- b) Írjon egy C/C++ programot, amely a billentyűzetről beolvas egy `n` ( $n < 10000$ ) természetes számot, és a `sdiv` alprogram hivatkozásait használva ellenőrzi, hogy az `n` osztóinak összege prímszám-e. Ha igen, akkor a program kiírja a **DA** üzenetet, ellenkező esetben a **NU** üzenetet.

**Példa:** ha `n=206`, akkor a kiírt üzenet: **NU** ( $1+2+103+206=312$ , és 312 nem prímszám). **(5p.)**