

30-31. ÓRA**További összetett programozási tételek****Szétválogatás tétele**

Adott n elemű $A[]$ és egy ezen értelmezett T tulajdonság. Feladatunk a tulajdonságnak megfelelő elemek kiválogatása az $E[]$ -be, a tulajdonságnak meg nem felelő elemek különválogatása egy $F[]$ -be. Legyen m az $E[]$ elemszáma, o pedig az $F[]$ elemeinek száma.

A megoldás szöveges algoritmus leíró módon a következő:

ELJÁRÁS SZÉTVÁLOGHATÁS

```

i := 1
M := 0
O := 0
CIKLUS AMÍG i<=N
    HA A[i] eleme T tulajdonságú, AKKOR
        M := M+1
        E[M] := A[i]
    EGYÉBKÉNT
        O := O+1
        F[O] = A[i]
    HA VÉGE
        I := I+1
CIKLUS VÉGE

KI: M, E[] elemei
KI: O, F[] elemei

```

ELJÁRÁS VÉGE

A tétel ANSI C megvalósítása:

```

#include <stdio.h>
#include "MyLIB.h"

#define MAXDB 20
#define MIN 20.0
#define MAX 40.0

int main(void) {
    int i, m, n, o;
    float a[MAXDB], e[MAXDB], f[MAXDB];
    float t;

    // A[] elemeinek generálása
    n = rndint(1, MAXDB);
    rndfltarr(MIN, MAX, n, a[]);

    // T tulajdonság (felső határérték)
    t = rndflt(MIN, MAX);

    // szétválogatás tétele
    m = o = 0;
    for (i=0; i<n; i++) {
        if (a[i]<=t) { e[m++] = a[i]; }
        else { f[o++] = a[i]; }
    }

    // E[] elemek kiírása
    printf("\nMegfelelo tetelek (%d db): ", m);
    for (i=0; i<m; i++) {
        printf("%s%.2f", ((i==0)?(""):(", ")), e[i]);
    }
}

```

```

// F[] elemek kiírása
printf("\nNem megfelelo tetelek (%d db): ", o);
for (i=0; i<o; i++) {
    printf("%s%.2f", ((i==0)?(""):(", ")), f[i]);
}

return (0);
}

```

Metszet képzés tétele

Adott két tömb (A[] és B[]), amelyek **n**, illetve **m** eleműek. Feladatunk, hogy az E[]-be válogassuk ki azon elemeket, amelyek mindkét tömbben szerepelnek. Az eredménytömb elemeinek száma legyen **o**.

A megoldás szöveges algoritmus leíró módon a következő:

ELJÁRÁS METSZET

```

i := j := 1
o := 0
CIKLUS AMÍG i<=N
    CIKLUS AMÍG j<=M ÉS A[i]!=B[j]
        j := j+1
    CIKLUS VÉGE
    HA j<=M AKKOR
        o := o+1
        E[o] = A[i]
    HA VÉGE
        i := i+1
CIKLUS VÉGE

KI: o, E[] elemei

```

ELJÁRÁS VÉGE

A tétel ANSI C megvalósítása:

```

#include <stdio.h>
#include "MyLIB.h"

#define MAXDB 10
#define MIN 20
#define MAX 40

int main(void) {
int i, j, m, n, o;
int a[MAXDB], b[MAXDB], e[MAXDB];

// A[] elemszámának, elemeinek generálása és kiírása
n = rndint(1, MAXDB);
rndintarrunq(MIN, MAX, a, n);

printf("\nA[] elemei (%d db): ", n);
for (i=0; i<n; i++) {
    printf("%s%d", ((i==0)?(""):(", ")), a[i]);
}

// B[] elemszámának, elemeinek generálása és kiírása
m = rndint(1, MAXDB);
rndintarrunq(MIN, MAX, b, m);

printf("\nB[] elemei (%d db): ", m);
for (i=0; i<m; i++) {
    printf("%s%d", ((i==0)?(""):(", ")), b[i]);
}

// metszetképzés tétele
o = 0;
for (i=0; i<n; i++) {
    for (j=0; j<m && a[i]!=b[j]; j++) { ; }
    if (j<m) { e[o++] = a[i]; }
}

```

```
printf("\n\nKozos elemek (%d db): ", o);  
for (i=0; i<o; i++) {  
    printf("%s%d", ((i==0)?(""):(", ")), e[i]);  
}  
  
return (0);}
```

Házi feladat

ANSI C forrásprogramok elkészítése és futtatása.
Eredmény képernyők e-mail-ben megküldése.