

8.hét

Az óra témája

Az órán pár egyszerű algoritmus kerül bemutatásra (buborékrendezés, rekurzió).

Mintaprogram

A következő példaprogramban egy tömböt feltöltünk random értékekkel, majd ennek a tartalmát buborékrendezéssel növekvő sorrendbe rendezzük.

Forráskód

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define ARRAY_SIZE  (100)
#define MAX_NUM      (1000)

void bubble( int* arr, int size );

int main()
{
    int t[ARRAY_SIZE], i = 0, size;
    int t_bubble[ARRAY_SIZE];
    srand(time(NULL));

    size = sizeof(t);

    printf( "A rendezetlen tomb:\n" );

    while( i < ARRAY_SIZE)
    {
        t[i] = rand() % MAX_NUM;
        printf( "t[%2d] = %4d\n", i, t[i] );
        i++;
    }
    printf("\n");

    memcpy( &t_bubble[0], &t[0], size);
    bubble( &t_bubble[0], ARRAY_SIZE);

    printf("Buborek rendezes utan:\n");

    i = 0;
    while( i < ARRAY_SIZE)
    {
        printf( "t_bubble[%2d] = %4d\n", i, t_bubble[i] );
        i++;
    }
}
```

```

        return 0;
    }

void bubble( int* arr, int size )
{
    int i, j, l, tmp;
    for(i = 0; i < (size-1); i++)
    {
        l = 0;
        for(j = 0; j < (size-1-i); j++)
        {
            if(arr[j] > arr[j+1])
            {
                tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
                l = 1;
            }
        }
        if(l == 0)
        {
            break;
        }
    }
}

```

Mintaprogram

A következő példa a Fibonacci-számokat számolja ki (<https://hu.wikipedia.org/wiki/Fibonacci-sz%C3%A1mok>) rekurzív (önmagát hívó) módon.

```

#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

#define SIZE    (10)

unsigned int fibonacci(int x)
{
    //    if(x <= 0) return -1;
    if(x == 1 || x == 2)
        return 1;
    else
        return fibonacci(x - 1) + fibonacci(x - 2);
}

int main()
{
    unsigned int i, t[SIZE];
    for (i = 1; i <= SIZE; i++)

```

```
{
    t[i-1] = fibonacci(i);
}
for (i = 0; i < SIZE; i++)
{
    printf( "%2d.\t%10d\n", i, t[i] );
}
return 0;
}
```