

[< Programozási tételek](#)

Programozási tételek C# megvalósításban

- **Szerző:** Sallai András
- Copyright © Sallai András, 2011, 2016
- Licenc: GNU Free Documentation License 1.3
- Web: <http://szit.hu>

Összegzés

[osszegzes.cs](#)

```
using System;
class Hello
{
    static void Main() {

        int[] tomb = {8, 9, 5, 4, 1};
        int n = 5;

        Console.WriteLine("Összegzés tétel");
        int osszeg = 0;
        for (int i = 0; i < n; i++)
            osszeg = osszeg + tomb[i];
        Console.WriteLine("Összeg: " + osszeg);

    }
}
```

Megszámolás

[megszamolas.cs](#)

```
/* Szeretnénk megszámlálni az 5-nél kisebb számokat */
using System;
class Program
{
    static void Main()
    {
        int[] t = {9, 7, 3, 5, 4, 2, 6};
        int n = t.Length;
        int c = 0;
```

```
        for(int i=0; i<n; i++)
            if(t[i] < 5)
                c++;

        Console.WriteLine("5-nél kisebb számok darabszáma: {0}", c);
    }
}
```

Eldöntés

[eldontes.cs](#)

```
/* El kell döntenünk, hogy egy adott elem szerepel-e egy tömbben */

using System;
class Program
{
    static void Main()
    {
        int[] t = {9, 7, 3, 5, 4, 2, 6};
        int n = t.Length;
        int ker = 5; //Keresett érték

        bool van = false;
        for(int i=0; i<n; i++)
            if(t[i] == ker)
                van = true;

        Console.WriteLine("Igaz-e, hogy van 5-ös a tömbben?: {0}",
van);
    }
}
```

[eldontes2.cs](#)

```
/* Benne van-e a keresett szám hatékonyabban */
using System;

class Program
{
    static void Main()
    {
        int[] t = {9, 7, 3, 5, 4, 2, 6};
        int n = t.Length;
        int ker = 5; //Keresett érték
```

```
int i = 0;
while(i<n && t[i]!=ker)
    i++;

if(i<n)
    Console.WriteLine("Benne van ");
else
    Console.WriteLine("Nincs benne");
}
```

Kiválasztás

[kivalasztas.cs](#)

```
/* Szeretnénk kiválasztani, hányadik helyen szerepel egy adott elem egy
tömbben */
using System;
class Program
{
    static void Main()
    {
        int[] t = {9, 7, 3, 5, 4, 2, 6};
        int n = t.Length;
        int ker = 5; //Keresett érték

        int i = 0;
        while(t[i] != ker)
            i++;

        Console.WriteLine("Az 5-ös indexe: {0}", i);
    }
}
```

Keresés

[kereses.cs](#)

```
/* Benne van-e a keresett szám, és hányadik helyen van */
using System;

class Program
{
```

```
static void Main()
{
    int[] t = {9, 7, 3, 5, 4, 2, 6};
    int n = t.Length;
    int ker = 5; //Keresett érték

    int i = 0;
    while(i<n && t[i]!=ker)
        i++;

    if(i<n)
        Console.WriteLine("Indexe: {0}", i);
    else
        Console.WriteLine("Nincs benne");
}
```

Kiválogatás

[kivalogatas.cs](#)

```
/* Szeretnénk kiválogatni a adott feltételek szerint egy tömb elemeit
*/
using System;

class Program
{
    static void Main()
    {
        int[] a = {9, 7, 3, 5, 4, 2, 6};
        int n = a.Length;
        int[] b = new int[n];

        int j = 0;
        for(int i=0; i<n; i++)
            if(a[i] < 5) //Az 5-nél kisebb számokat válogatjuk
            {
                b[j] = a[i];
                j++;
            }

        Console.WriteLine("Eredeti:");
        for(int i=0; i<n; i++)
            Console.Write("{0} ", a[i]);
        Console.WriteLine();

        Console.WriteLine("Kiválogatott:");
    }
}
```

```
        for(int i=0; i<j; i++)
            Console.Write("{0} ", b[i]);
        Console.WriteLine();
    }
}
```

Szétválogatás

[szetvalogatás.cs](#)

```
/* Szeretnénk kiválogatni a adott feltételek szerint egy tömb elemeit
*/
using System;

class Program
{
    static void Main()
    {
        int[] a = {9, 7, 3, 5, 4, 2, 6};
        int n = a.Length;
        int[] b = new int[n];
        int[] c = new int[n];

        int j = 0;
        int k = 0;
        for(int i=0; i<n; i++)
            if(a[i] < 5) //Az 5-nél kisebb számokat válogatjuk
            {
                b[j] = a[i];
                j++;
            }
            else
            {
                c[k] = a[i];
                k++;
            }

        Console.WriteLine("Eredeti:");
        for(int i=0; i<n; i++)
            Console.Write("{0} ", a[i]);
        Console.WriteLine();

        Console.WriteLine("Kiválogatott b:");
        for(int i=0; i<j; i++)
            Console.Write("{0} ", b[i]);
        Console.WriteLine();
    }
}
```

```
        Console.WriteLine("Kiválogatott c:");  
        for(int i=0; i<k; i++)  
            Console.Write("{0} ", c[i]);  
        Console.WriteLine();  
    }  
}
```

Metszet

[metszet.cs](#)

```
using System;  
class Program  
{  
    static void Main()  
    {  
        int[] a = {5, 9, 3, 4, 7 };  
        int[] b = {6, 5, 7, 8, 15, 20 };  
        int[] c = new int[10];  
        int n = 5, m = 6, o; //Tömbök mérete  
        int i, j, k; //Ciklusváltozók, illetve tömbindexek  
  
        k = 0;  
        for (i=0; i<n; i++)  
        {  
            j = 0;  
            while(j<m && b[j] != a[i])  
                j++;  
            if (j<m)  
            {  
                c[k] = a[i];  
                k++;  
            }  
        }  
  
        o = k; //Harmadik azaz a "c" tömb mérete  
  
        /* Tömbök kiíratása */  
        for (i=0; i<n; i++)  
            Console.Write(a[i] + " ");  
        Console.WriteLine();  
        for (j=0; j<m; j++)  
            Console.Write(b[j] + " ");  
        Console.WriteLine();  
        for (k=0; k<o; k++)  
            Console.Write(c[k] + " ");  
    }  
}
```

```
    Console.WriteLine();  
}  
}
```

Unió tétel

Két tömb elemeit egy harmadik tömbbe tároljuk.

[unio.cs](#)

```
using System;  
  
class Program  
{  
    static void Main()  
    {  
        Console.WriteLine("Unió tétel");  
        int[] a = {3, 5, 8, 4};  
        int[] b = {2, 1, 7, 9};  
        int[] c = new int[18];  
  
        int i, j, k;  
        int n=a.Length, m=b.Length;  
  
        //Unió tétel  
        for(i=0; i<n; i++)  
            c[i] = a[i];  
        k=n;  
        for(j=0; j<m; j++)  
        {  
            i = 0;  
            while(i<n && b[j] != a[i])  
                i++;  
            if(i >= n)  
            {  
                c[k] = b[j];  
                k++;  
            }  
        }  
  
        //Az a tömb kiíratása  
        for(i=0; i<n; i++)  
            Console.Write(a[i] + " ");  
        Console.WriteLine();  
  
        //A b tömb kiíratása  
        for(i=0; i<m; i++)
```

```
        Console.Write(b[i] + " ");  
        Console.WriteLine();  
  
        //A c eredménytömb kiíratása  
        for(i=0; i<k; i++)  
            Console.Write(c[i] + " ");  
        Console.WriteLine();  
    }  
}
```

Maximum kiválasztás

[maxkiv.cs](#)

```
using System;  
class Program  
{  
    public static void Main()  
    {  
        int[] t = {4, 3, 9, 7, 2, 5};  
        int n = t.Length; //Az n a tömb mérete  
        int max;  
  
        //Maximum kiválasztás tétele  
        max = t[0];  
        for(int i=0; i<n; i++)  
            if(t[i]>max)  
                max = t[i];  
  
        Console.WriteLine("A legnagyobb elem: {0}", max);  
    }  
}
```

Minimum kiválasztás

[minkiv.cs](#)

```
using System;  
class Program  
{  
    public static void Main()  
    {  
        int[] t = {4, 3, 9, 7, 2, 5};
```



```
int n = t.Length; //Az n a tömb mérete
int min;

//Maximum kiválasztás tétele
min = t[0];
for(int i=1; i<n; i++)
    if(t[i]<min)
        min = t[i];

Console.WriteLine("A legkisebb elem: {0}", min);
}
```

Rendezések

Buborék rendezés

[buborek.cs](#)

```
using System;

class Program
{
    static void Main()
    {
        int[] t = new int[] {5, 4, 9, 3, 7};
        int n = t.Length;

        //Kiíratás rendezés előtt
        for(int i=0; i<n; i++)
            Console.Write("{0} ", t[i]);
        Console.WriteLine();

        //Buborék rendezés
        for(int i=n-1; i>0; i--)
            for(int j=0; j<i; j++)
                if(t[j]>t[j+1])
                {
                    int tmp = t[j+1];
                    t[j+1] = t[j];
                    t[j] = tmp;
                }

        //Kiíratás rendezés után
        for(int i=0; i<n; i++)
            Console.Write("{0} ", t[i]);
    }
}
```

```
        Console.WriteLine();  
    }  
}
```

Cserés rendezés

[cseresrendezes.cs](#)

```
using System;  
  
class Program  
{  
    static void Main()  
    {  
        int[] t = {22, 5, 4, 33, 9, 3, 7,15,20};  
        int n = t.Length;  
        //Kiíratás rendezés előtt  
        for(int i=0; i<n; i++)  
            Console.Write("{0} ", t[i]);  
        Console.WriteLine();  
  
        //Cserés rendezés  
        for(int i=0; i<n -1; i++)  
            for(int j=i+1; j<n;j++)  
                if(t[i]>t[j])  
                {  
                    int swap = t[j];  
                    t[j] = t[i];  
                    t[i] = swap;  
                }  
  
        //Kiíratás rendezés után  
        for(int i=0; i<n; i++)  
            Console.Write("{0} ", t[i]);  
        Console.WriteLine();  
    }  
}
```

Rendezés maximumkiválasztással

[maxkivrend.cs](#)

```
using System;
class Program
{
    public static void Main()
    {
        int[] t = { 8, 3, 9, 1, 5, 2, 7};
        int n = t.Length;

        for(int i=0; i<n; i++)
            Console.Write(t[i] + " ");
        Console.WriteLine();

        for(int i=n-1; i>0; i--)
        {
            int max = i;
            for(int j=0; j<=i; j++)
                if(t[j]>t[max])
                    max = j;
            int swap = t[i];
            t[i] = t[max];
            t[max] = swap;
        }

        for(int i=0; i<n; i++)
            Console.Write(t[i] + " ");
        Console.WriteLine();
    }
}
```

Beszúrásos rendezés

[beszurend.cs](#)

```
using System;

class Program
{
    public static void Main()
    {
        int[] t = {5, 2, 4, 3, 9, 7};

        for(int i=1; i<t.Length; i++)
        {
            int kulcs = t[i];
            int j = i - 1;
            while(j>=0 && t[j]>kulcs)
```

```
        {
            t[j+1] = t[j];
            j = j - 1;
        }
        t[j+1] = kulcs;

    }

    for(int i=0; i<t.Length; i++)
        Console.Write(t[i] + " ");
    Console.WriteLine();
}
}
```

Gyorsrendezés

A rendezendő számokat két részre bontjuk, majd ezeket a részeket rekurzívan, gyorsrendezéssel rendezzük.

[gyorsrend.cs](#)

```
using System;
class Program
{
    static int[] t = { 8, 3, 9, 1, 5, 2, 7};
    static void gyorsrendezes(int also, int felso)
    {
        int i=also, j=felso;
        int kozep = t[(felso+also)/2];
        while(also<=felso)
        {
            while(also<j && t[also]<kozep)
                also++;
            while(felso>i && t[felso]>kozep)
                felso--;
            if(also<=felso)
            {
                int tmp = t[also];
                t[also] = t[felso];
                t[felso] = tmp;
                ++also;
                --felso;
            }
        }
        if(also<j) gyorsrendezes(also, j);
    }
}
```

```
        if(i<felso) gyorsrendezes(i, felso);

    }

    static void Main()
    {

        int n = t.Length;

        for(int i=0; i<n; i++)
            Console.Write(t[i] + " ");
        Console.WriteLine();

        gyorsrendezes(0,6);

        for(int i=0; i<n; i++)
            Console.Write(t[i] + " ");
        Console.WriteLine();

    }
}
```

Shell rendezés

[Shellrendezes.cs](#)

```
using System;

class Program
{
    static void Main()
    {
        int[] t = new int[] {22, 5, 4, 33, 9, 3, 7, 15, 20};
        int[] h = { 5, 3, 1};
        int n = t.Length;

        //Kiíratás rendezés előtt
        for(int i=0; i<n; i++)
            Console.Write("{0} ", t[i]);
        Console.WriteLine();

        //Shell rendezés
        for(int k=0; k<h.Length; k++)
        {
            int lepes = h[k];
            for(int j=lepes; j<n; j++)
            {
                int i = j - lepes;
                int x = t[j];
```

```
        while(i >= 0 && t[i] > x)
        {
            t[i+lepes] = t[i];
            i = i - lepes;
        }
        t[i+lepes] = x;
    }

    //Kiíratás rendezés után
    for(int i=0; i<n; i++)
        Console.Write("{0} ", t[i]);
    Console.WriteLine();
}
```

Egyéb példák

Cserés rendezés ListBox-ban

```
private void Cseresrendezes()
{
    List<string> idlist = new List<string>();

    foreach(string s in downlistbox.Items)
        idlist.Add(s);

    int n = idlist.Count;

    //Cserés rendezés
    for(int i=0; i<n-1; i++)
        for(int j=i+1; j<n; j++)
        {
            if(Convert.ToInt32(idlist[i].Split('.')[0]) > Convert.ToInt32(idlist[j].Split(
                '.')[0]))
            {
                string swap = idlist[j];
                idlist[j] = idlist[i];
                idlist[i] = swap;
            }
        }

    downlistbox.Items.Clear();

    downlistbox.Items.AddRange(idlist.ToArray());
}
```

```
}
```

Erre azért lehet, szükség, mert más szempontok alapján akarok listázni. Például ilyen elemeket tartalmaz:

- 1.) Valami
- 2.) Másik valami
- 10.) Harmadik
- 11.) Negyedik

Ebben az esetben a szimpla ábécé szerint rendez, amely másodikat a végére teszi, mivel az azt következők 1-el kezdődnek.

From:
<http://szit.hu/> - SzitWiki

Permanent link:
http://szit.hu/doku.php?id=oktatas:programoz%C3%A1s:programoz%C3%A1si_t%C3%A9telek:csharp_megval%C3%B3s%C3%ADt%C3%A1s

Last update: 2017/10/02 20:29

