

Laborator 5

Tablouri

Tablouri unidimensionale

- Limbajul C# tratează tablourile într-o manieră nouă față de alte limbaje (Pascal, C/C++).

La declararea unui tablou, se creează o instanță a clasei .NET, **System.Array**. Compilatorul va traduce operațiile asupra tablourilor, apelând metode ale **System.Array**.

Declararea unui tablou unidimensional se face astfel:

```
Tip[] nume;
```

Prin această declarație nu se alocă și spațiu pentru memorare. Pentru aceasta, tabloul trebuie **instanțiat**:

```
nume = new Tip[NumarElemente];
```

Se pot face în același timp operațiile de declarare, instanțiere și inițializare:

Exemplu:

```
int[] v = new int[] {1,2,3};
```

Sau

```
int[] v = {1,2,3};
```

Exemplul 41: Crearea, sortarea și afișarea unui vector:

```
int[] v = new int[5] { 10, 2, 4, 8, 6 };
Array.Sort(v); //sortarea crescatoare a vectorului v

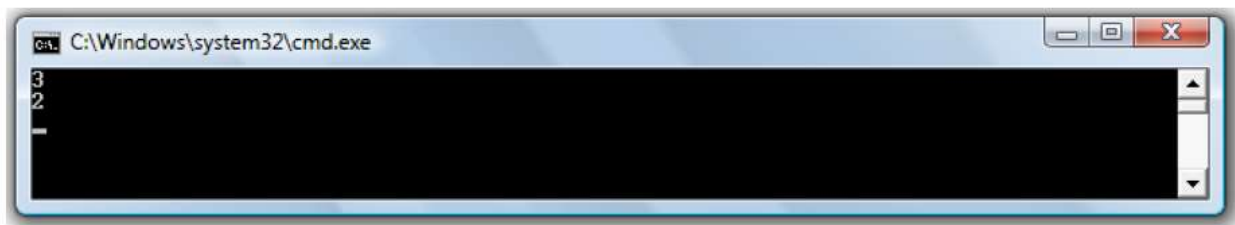
for (int i = 0; i < v.Length; i++) //afisarea vectorului v
    Console.WriteLine("{0,3}", v[i]);
```

Afișarea se poate face și cu ajutorul lui **foreach**:

```
foreach (int i in v)
    Console.WriteLine("{0,3}", i);
```

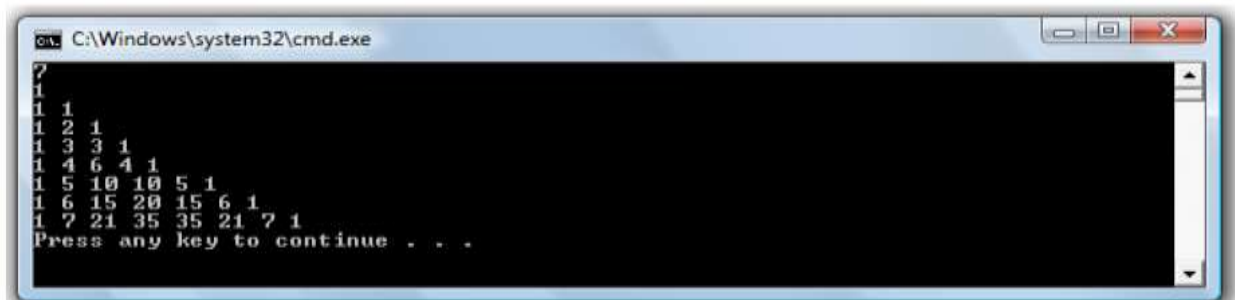
Exemplul 42: Să se afișeze numărul de elemente de pe a doua linie a tabloului și numărul total de linii.

```
using System;
namespace Exemplul_42
{
    class Program
    {
        static void Main(string[] args)
        {
            int[,] tab = { { 1, 2, 3 }, { 4, 5, 6 } };
            // Afișarea numărului de elemente ale
            // lui tab de pe linia a 2-a.
            // Reamintim ca prima linie are numărul de ordine 0
            Console.WriteLine(tab.GetLength(1));
            // Afișarea numărului de linii a tabloului tab
            Console.WriteLine(tab.Rank);
            Console.ReadLine();
        }
    }
}
```



Exemplul 43: Să se afișeze primele $n+1$ linii din triunghiul lui PASCAL($n \leq 20$).

```
using System;
namespace Exemplul_43
{
    class Program
    {
        static void Main()
        {
            int n, i, j; int[] p, q;
            n = Convert.ToInt32(Console.ReadLine());
            p = new int[n + 1]; q = new int[n + 1];
            p[0] = 1;
            for (i = 1; i <= n + 1; i++)
            {
                q[0] = 1; q[i - 1] = 1;
                for (j = 1; j <= i - 2; j++)
                    q[j] = p[j - 1] + p[j];
                for (j = 0; j <= i - 1; j++)
                {
                    Console.Write(q[j] + " ");
                    p[j] = q[j];
                }
                Console.WriteLine();
            }
        }
    }
}
```

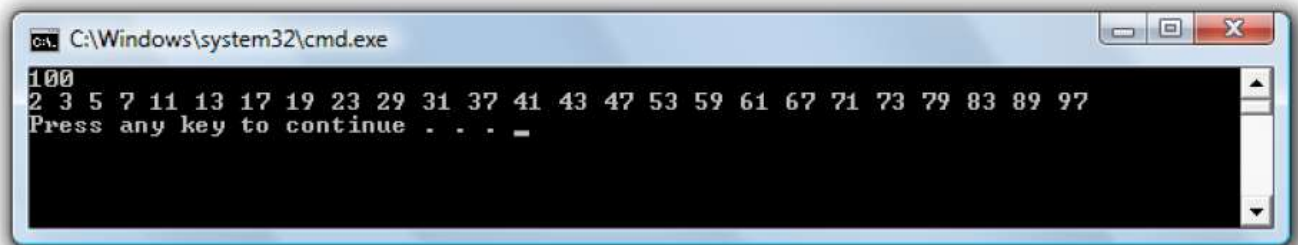


Exemplul 44: **Ciurul lui Eratostene.** Pentru un număr natural n dat se afișează toate numerele prime mai mici decât n . Selectarea numerelor prime se face folosind ciurul lui Eratostene Ciurul lui Eratostene presupune formarea unui șir din numerele 2, 3, 4, ..., $n-1$, n . Pentru a obține acest șir „tăiem” mai întâi toți multiplii lui 2, apoi ai lui 3 ș.a.m.d. În final rămân numai numerele prime din intervalul $[2, n]$. Noțiunea de „tăiere” a unui element va însemna, în acest caz, atribuirea valorii zero pentru acel element.

```

using System;
namespace Exemplul_44
{
    class Program
    {
        static void Main()
        {
            int n, i, j, k;
            int[] c;
            n = Convert.ToInt32(Console.ReadLine());
            c = new int[n + 1];
            for (i = 2; i <= n; i++)
                c[i] = i;
            i = 2;
            while (i <= n / 2) //cel mai mare divizor propriu al unui numar
            este<=jumatatea sa
            {
                if (c[i] != 0)
                {
                    j = 2 * i;
                    while (j <= n)
                    {
                        if (c[j] != 0) c[j] = 0;
                        j += i;
                    }
                }
                i++;
            }
            for (i = 2; i <= n; i++)
                if (c[i] != 0)
                    Console.Write(c[i] + " ");
            Console.WriteLine();
        }
    }
}

```



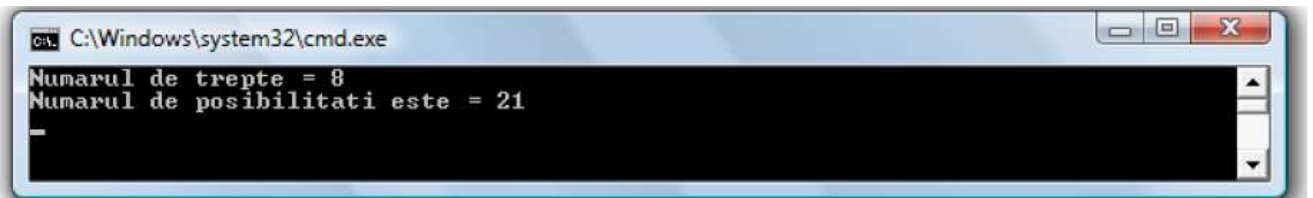
Exemplul 45: Ionel urcă în fiecare zi n trepte ($n < 40$) până la apartamentul în care locuiește. El poate urca pășind pe treapta următoare sau sărind peste o treaptă. În câte moduri poate urca Ionel cele n trepte?

Dacă notăm cu $f[i]$ numărul de moduri în care poate urca copilul i trepte, observăm că există 2 moduri prin care acesta poate ajunge la treapta i : de la treapta $i-1$ sau de la treapta $i-2$. Pentru a determina numărul de moduri, vom însuma în câte moduri poate ajunge pe treapta $i-1$ cu numărul de modalități de a ajunge pe treapta $i-2$, deci $f[i] = f[i-1] + f[i-2]$.

```

using System;
namespace Exemplul_45
{
    class Program
    {
        static void Main()
        {
            int n, i;
            int[] f;
            Console.Write("Numarul de trepte = ");
            n = Convert.ToInt32(Console.ReadLine());
            f = new int[n + 1];
            f[1] = f[2] = 1;
            for (i = 3; i <= n; i++) f[i] = f[i - 1] + f[i - 2];
            Console.WriteLine("Numarul de posibilitati este = 
{0}", f[n].ToString());
            Console.ReadLine();
        }
    }
}

```



Exemplul 46: Să se determine valoarea elementului maxim dintr-un tablou unidimensional, precum și frecvența sa de apariție

```

using System;
namespace Exemplul_46
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, i, max, f;
            int[] a;
            Console.Write("Dati dimensiunea tabloului : ");
            n = Convert.ToInt32(Console.ReadLine());
            a = new int[n + 1];
            for (i = 0; i < n; i++)
            {
                Console.Write(" a[ {0} ] = ", i + 1);
                a[i] = Convert.ToInt32(Console.ReadLine());
            }

            max = a[0];
            f = 1;
            for (i = 1; i < n; i++)
            {
                if (a[i] > max)
                {
                    max = a[i];
                    f = 1;
                }
                else
                {
                    if (a[i] == max) f++;
                }
            }
            Console.WriteLine("Maximul din tablou este {0} cu frecventa {1}
", max, f);
        }
    }
}

```



```
C:\Windows\system32\cmd.exe
Dati dimensiunea tabloului : 9
a[ 1 ] = 1
a[ 2 ] = 6
a[ 3 ] = 9
a[ 4 ] = 1
a[ 5 ] = 9
a[ 6 ] = 7
a[ 7 ] = 4
a[ 8 ] = 9
a[ 9 ] = 3
Maximul din tablou este 9 cu frecventa 3
Press any key to continue . . . _
```

Exemplul 47: Operații cu elementele unui vector: citire, afișare, eliminare elemente de valoare 0, inserare după fiecare valoare a celei mai apropiate puteri ale lui 2 (dacă cele două puteri sunt la aceeași distanță față de număr se va insera cea mai mică dintre cele doua puteri)

```
using System;

namespace Exemplul_47
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, i, j, k = 0;
            int[] a;
            Console.WriteLine("Dati dimensiunea tabloului : ");
            n = Convert.ToInt32(Console.ReadLine());
            a = new int[2 * n + 1];
            Console.WriteLine("Citire tablou : ");
            for (i = 0; i < n; i++)
            {
                Console.Write(" a[ {0} ] = ", i + 1);
                a[i] = Convert.ToInt32(Console.ReadLine());
            }
            Console.WriteLine("Afișare tablou : ");
            for (i = 0; i < n; i++)
                Console.Write("{0} ", a[i]);
            Console.WriteLine();
            // stergere valori nule
            i = 0;
            while (a[i] != 0 && i < n) i++;
            while (i < n)
            {
                if (a[i] == 0)
                {
                    for (j = i; j < n && a[j] == 0; j++) ;
                    a[i++] = a[j];
                    a[j] = 0;
                    k++;
                }
                else i++;
            }
            while (a[n - 1] == 0 && n > 0) n--;
            Console.WriteLine("Afișare tablou fara valori nule : ");
            for (i = 0; i < n; i++)
                Console.Write("{0} ", a[i]);
            Console.WriteLine();
            // inserare valori
            for (i = 0; i < n; i += 2)
            {
                for (j = n; j > i; j--)
                    a[j] = a[j - 1];
                a[i + 1] = putere(a[i]);
                n++;
            }
            Console.WriteLine("Afișare tablou dupa inserare puteri ale lui 2 : ");
            for (i = 0; i < n; i++)
                Console.Write("{0} ", a[i]);
            Console.WriteLine();
        }
    }
}
```

```

static int putere(int x)
{
    int p = 1, q;
    while (p <= x) p *= 2;
    q = p / 2;
    if (x - q <= p - x) return q;
    else return p;
}
}

```

```

C:\Windows\system32\cmd.exe
Dati dimensiunea tabloului : 8
Citire tablou :
a[1] = 4
a[2] = 8
a[3] = 0
a[4] = 2
a[5] = 6
a[6] = 0
a[7] = 1
a[8] = 9
Afisare tablou :
4 8 0 2 6 0 1 9
Afisare tablou fara valori nule :
4 8 2 6 1 9
Afisare tablou dupa inlocuire puteri ale lui 2 :
4 4 8 8 2 2 6 4
Press any key to continue . . . . _

```