

Laborator 3

Instrucțiuni condiționale, de iterație și de control

Ne referim aici la instrucțiunile construite folosind cuvintele cheie: **if, else, do, while, switch, case, default, for, foreach, in, break, continue, goto.**

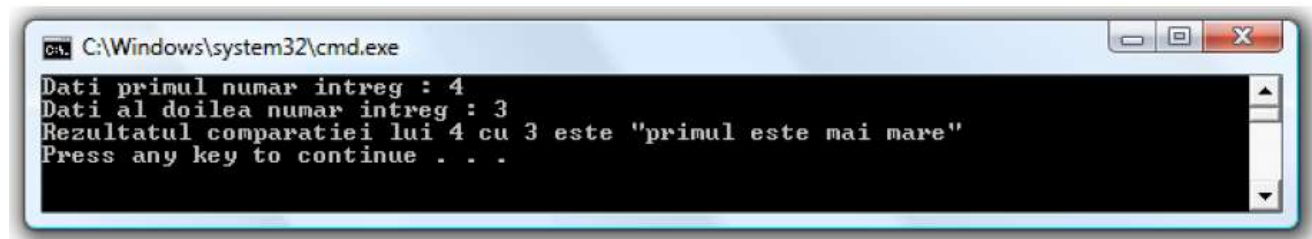
Instrucțiunea if

Instrucțiunea if are sintaxa:

```
if (conditie)
    Instructiuni_A;
else
    Instructiuni_B;
```

Exemplul 23: Citindu-se două numere întregi, să se decidă care dintre ele este mai mare

```
using System;
namespace Exemplul_23
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b;
            string rezultat;
            Console.Write("Dati primul numar intreg : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Dati al doilea numar intreg : ");
            b = Convert.ToInt32(Console.ReadLine());
            if (a > b) rezultat = "primul este mai mare";
            else
                if (a < b) rezultat = "primul este mai mic";
                else rezultat = "numere egale";
            Console.WriteLine("Rezultatul comparatiei lui {0} cu {1} este
\\\"{2}\\\" ", a, b, rezultat);
        }
    }
}
```



Exemplul 24: Să se verifice dacă 3 puncte din plan M_1 , M_2 și M_3 , date prin coordonatele lor întregi, sunt coliniare.

Punctele $M_1(x_1, y_1)$, $M_2(x_2, y_2)$, $M_3(x_3, y_3)$ sunt coliniare

$$\Leftrightarrow \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0$$

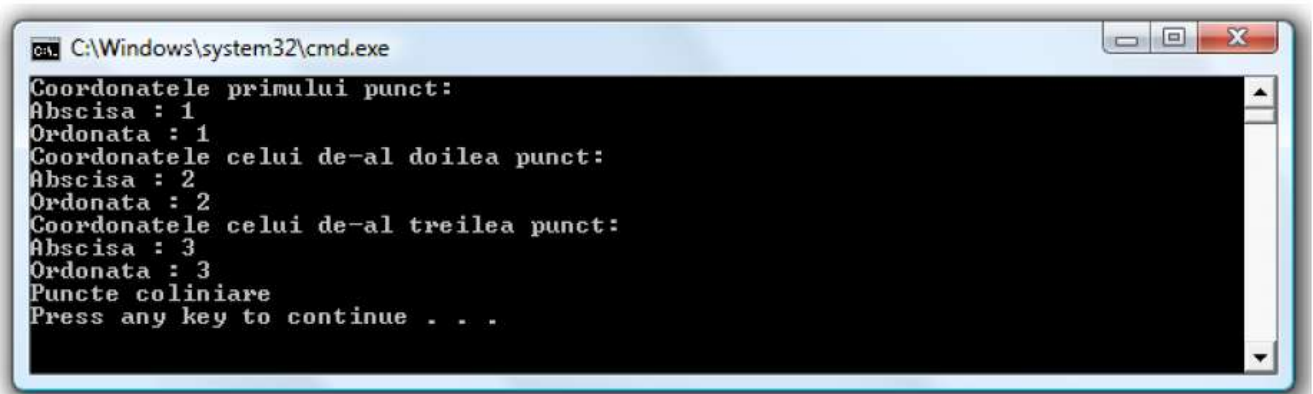
$$\Leftrightarrow E = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) = 0$$

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Exemplul_24
{
    class Program
    {
        static void Main(string[] args)
        {
            double x1, y1, x2, y2, x3, y3;
            Console.WriteLine("Coordonatele primului punct:");
            Console.Write("Abscisa : ");
            x1 = Convert.ToDouble(System.Console.ReadLine());
            Console.Write("Ordonata : ");
            y1 = Convert.ToDouble(System.Console.ReadLine());
            Console.WriteLine("Coordonatele celui de-al doilea punct:");
            Console.Write("Abscisa : ");
            x2 = Convert.ToDouble(System.Console.ReadLine());
            Console.Write("Ordonata : ");
            y2 = Convert.ToDouble(System.Console.ReadLine());
            Console.WriteLine("Coordonatele celui de-al treilea punct:");
            Console.Write("Abscisa : ");
            x3 = Convert.ToDouble(System.Console.ReadLine());
            Console.Write("Ordonata : ");
            y3 = Convert.ToDouble(System.Console.ReadLine());

            double E = (x2 - x1) * (y3 - y1) - (x3 - x1) * (y2 - y1);

            if (E == 0) Console.WriteLine("Puncte coliniare");
            else Console.WriteLine("Puncte necoliniare");
        }
    }
}
```



```
C:\Windows\system32\cmd.exe
Coordonatele primului punct:
Abscisa : 1
Ordonata : 1
Coordonatele celui de-al doilea punct:
Abscisa : 2
Ordonata : 2
Coordonatele celui de-al treilea punct:
Abscisa : 3
Ordonata : 3
Puncte coliniare
Press any key to continue . . .
```

Exemplul 25: Să se verifice dacă un număr întreg x este într-un interval dat $[a, b]$

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Exemplul_25
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, x;
            Console.WriteLine("Se citesc doua numere care vor reprezenta
capetele intervalului");
            Console.Write("Dati prima valoare : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Dati a doua valoare : ");
            b = Convert.ToInt32(Console.ReadLine());
            if (a > b)
            {
                x = a;
                a = b;
                b = x;
            } // interschimbarea valorilor pentru a avea intervalul [a, b]
            Console.Write("x = ");
            x = Convert.ToInt32(Console.ReadLine());
            if (x >= a && x <= b)
                Console.WriteLine("Numarul {0} este in intervalul [ {1}, {2}
]", x, a, b);
            else
                Console.WriteLine("Numarul {0} nu este in intervalul [ {1},
{2} ]", x, a, b);
        }
    }
}

```

```

C:\Windows\system32\cmd.exe
Se citesc doua numere care vor reprezenta capetele intervalului
Dati prima valoare : 4
Dati a doua valoare : -10
x = 3
Numarul 3 este in intervalul [ -10, 4 ]
Press any key to continue . . .

```

Instrucțiunea switch

În cazul instrucțiunii **switch** în C/C++, dacă la finalul instrucțiunilor dintr-o ramură **case** nu există **break**, se trece la următorul **case**. În C# se semnalează eroare. Există și aici posibilitatea de a face verificări multiple (în sensul de a trece la verificarea următoarei condiții din **case**) doar dacă **case**-ul nu conține instrucțiuni:

Instrucțiunea **switch** admite în C# variabilă de tip șir de caractere care să fie comparată cu șirurile de caractere din **case**-uri:

Exemplul 26: Programul următor afișează ultima cifră a numărului x_n , unde x și n sunt numere naturale citite de la tastatură.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Exemplul_26
{
    class Program
    {
        static void Main(string[] args)
        {
            int x, n, k, ux;
            Console.Write("Dati un numar natural ca baza a puterii : ");
            x = Convert.ToInt32(Console.ReadLine());
            Console.Write("Dati un numar natural ca exponent al puterii : ");
            n = Convert.ToInt32(Console.ReadLine());

            ux = x % 10; // ma intereseaza doar ultima cifra
            Console.WriteLine("Ultima cifra a lui {0} la puterea {1} este : ", x,
n);


            if (n == 0) Console.WriteLine(" 1 ");
            else
                switch (ux)
                {
                    case 0: Console.WriteLine(" 0 "); break;
                    case 1: Console.WriteLine(" 1 "); break;
                    case 2:
                        k = n % 4;
                        switch (k)
                        {
                            case 0: Console.WriteLine(" 6 "); break;
                            case 1: Console.WriteLine(" 2 "); break;
                            case 2: Console.WriteLine(" 4 "); break;
                            case 3: Console.WriteLine(" 8 "); break;
                        }
                        break;

```

```

                    case 3:
                        k = n % 4;
                        switch (k)
                        {
                            case 0: Console.WriteLine(" 1 "); break;
                            case 1: Console.WriteLine(" 3 "); break;
                            case 2: Console.WriteLine(" 9 "); break;
                            case 3: Console.WriteLine(" 7 "); break;
                        }
                        break;
                    case 4:
                        if (n % 2 == 0) Console.WriteLine(" 6 ");
                        else Console.WriteLine(" 4 ");
                        break;
                    case 5:
                        Console.WriteLine(" 5 ");
                        break;
                    case 6:
                        Console.WriteLine(" 6 ");
                        break;
                    case 7:
                        k = n % 4;
                        switch (k)
                        {
                            case 0: Console.WriteLine(" 1 "); break;
                            case 1: Console.WriteLine(" 7 "); break;
                            case 2: Console.WriteLine(" 9 "); break;
                            case 3: Console.WriteLine(" 3 "); break;
                        }
                        break;
                    case 8:
                        k = n % 4;
                        switch (k)
                        {
                            case 0: Console.WriteLine(" 6 "); break;
                            case 1: Console.WriteLine(" 8 "); break;
                            case 2: Console.WriteLine(" 4 "); break;
                            case 3: Console.WriteLine(" 2 "); break;
                        }
                        break;
                    case 9:
                        if (n % 2 == 0) Console.WriteLine(" 1 ");
                        else Console.WriteLine(" 9 ");
                        break;
                }
            }
        }
    }
}

```

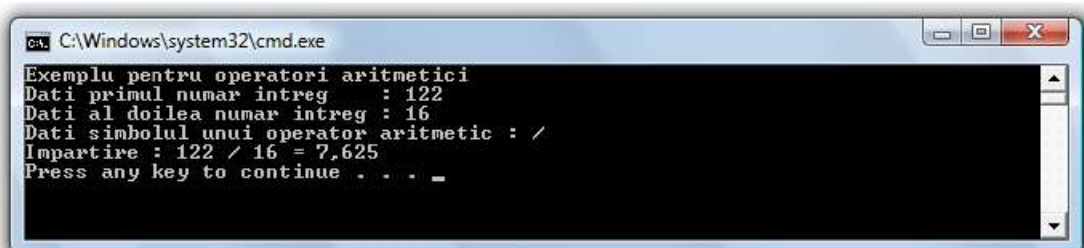



```
C:\Windows\system32\cmd.exe
Dati un numar natural ca baza a puterii : 273
Dati un numar natural ca exponent al puterii : 125
Ultima cifra a lui 273 la puterea 125 este : 3
Press any key to continue . . . _
```

Exemplul 27: Programul următor efectuează calculele corespunzătoare pentru două numere întregi și unul dintre semnele +, -, *, / , % introduse de la tastatură

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Exemplul_27
{
    class Program
    {
        static void Main(string[] args)
        {
            char op;
            int a, b;
            Console.WriteLine("Exemplu pentru operatori aritmetici");
            Console.WriteLine("Dati primul numar intreg : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Dati al doilea numar intreg : ");
            b = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Dati simbolul unui operator aritmetic : ");
            op = (char)Console.Read();
            switch (op)
            {
                case '+': Console.WriteLine("Adunare : {0} + {1} = {2}", a,
b, a + b);
                    break;
                case '-': Console.WriteLine("Scadere : {0} - {1} = {2}", a,
b, a - b);
                    break;
                case '*': Console.WriteLine("Inmultire : {0} * {1} = {2}", a,
b, a * b);
                    break;
                case '/': Console.WriteLine("Impartire : {0} / {1} = {2}", a,
b, (float)a / b);
                    break;
                case '%': Console.WriteLine("Modulo : {0} % {1} = {2}", a,
b, a % b);
                    break;
                default: Console.WriteLine("Simbolul nu reprezinta o operatie
aritmetica");
                    break;
            }
        }
    }
}
```



```
C:\Windows\system32\cmd.exe
Exemplu pentru operatori aritmetici
Dati primul numar intreg : 122
Dati al doilea numar intreg : 16
Dati simbolul unui operator aritmetic : /
Impartire : 122 / 16 = 7.625
Press any key to continue . . . _
```

Instrucțiunea while

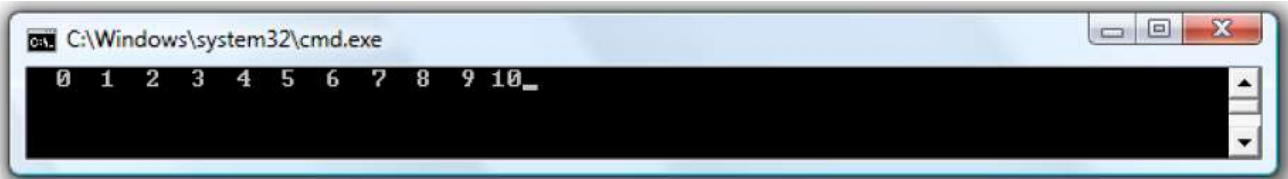
Instrucțiunea **while** are sintaxa:

```
while (conditie) Instructiuni;
```

Cât timp **conditie** este îndeplinită se execută **Instructiuni**.

Exemplul 28: Să se afișeze numerele întregi pozitive ≤ 10

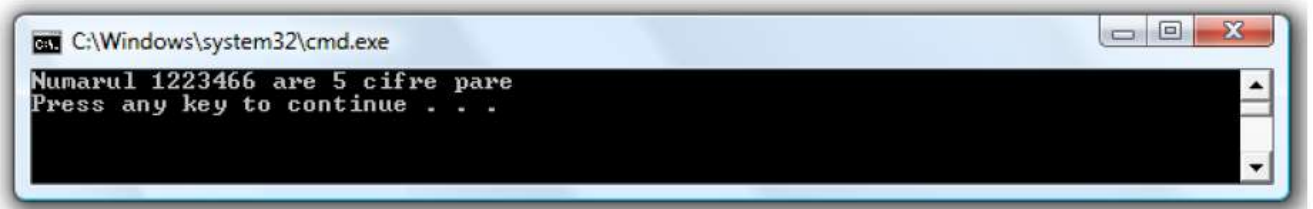
```
using System;
namespace Exemplul_28
{
    class Program
    {
        static void Main(string[] args)
        {
            int n = 0;
            while (n <= 10)
            {
                Console.Write("{0,3}", n);
                n++;
            }
            Console.ReadLine();
        }
    }
}
```



Exemplul 29: Programul de mai jos numără câte cifre pare are un număr natural:

```
using System;

namespace Exemplul_29
{
    class Program
    {
        static void Main(string[] args)
        {
            uint a = 1223466, b;
            b = CateCifrePare(a);
            Console.WriteLine("Numarul {0} are {1} cifre pare", a, b);
        }
        static uint CateCifrePare(uint a)
        {
            uint k = 0;
            if (a == 0) k = 1;
            while (a != 0)
            {
                if (a % 10 % 2 == 0) k++; // sau if(a % 2 == 0)
                // pentru ca a numar par daca si numai daca ultima cifra este
                para
                a = a / 10;
            }
            return k;
        }
    }
}
```



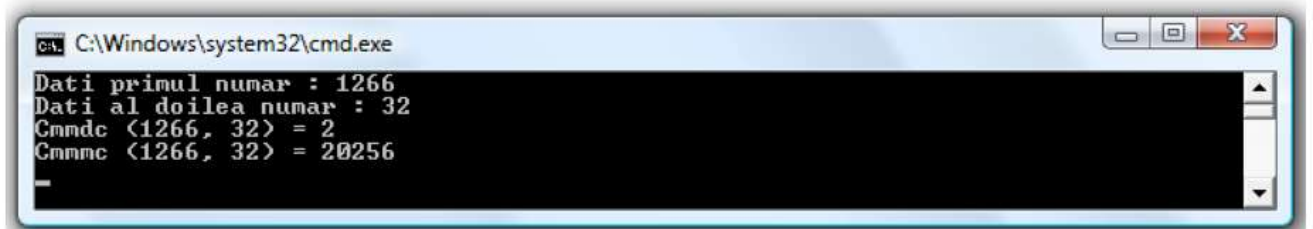
Exemplul 30: Să se calculeze cmmdc și cmmmc pentru două numere citite de la tastatură.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Exemplul_30
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b, r, x, y;
            Console.Write("Dati primul numar : ");
            a = Convert.ToInt32(Console.ReadLine());
```

```
            Console.Write("Dati al doilea numar : ");
            b = Convert.ToInt32(Console.ReadLine());

            x = a;
            y = b;
            r = x % y;
            while (r != 0)
            {
                x = y;
                y = r;
                r = x % y;
            }
            if (y != 1)
                Console.WriteLine("Cmmdc ({0}, {1}) = {2} ", a, b, y);
            else
                Console.WriteLine("{0} si {1} sunt prime intre ele ", a, b);
            Console.WriteLine("Cmmmc ({0}, {1}) = {2}", a, b, a / y * b);
            Console.ReadKey();
        }
    }
}
```



Exemplul 31: Dintr-un număr întreg pozitiv, citit de la tastatură, să se elimine cifra cea mai mică și să se afișeze numărul rezultat în urma acestei operații.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Exemplul_31
{
    class Program
    {
        static void Main(string[] args)
        {
            uint n, min, v;
            Console.WriteLine("Dati un numar intreg pozitiv : ");
            n = Convert.ToUInt32(Console.ReadLine());
            min = MinCifra(n);
            v = Valoare(n, min);
            Console.WriteLine("Eliminand cifra minima {0} din {1} obtinem
{2}", min, n, v);
        }
    }
}

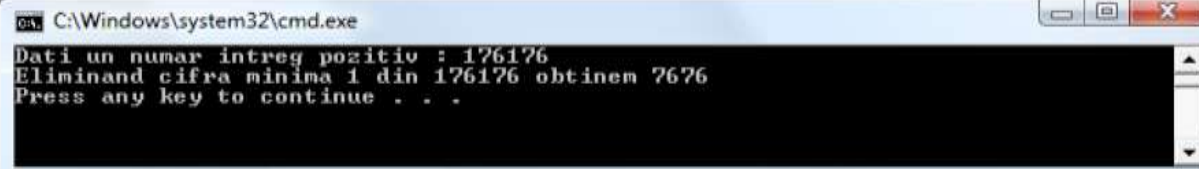
```

```

static uint MinCifra(uint x)
{
    uint min = 9;
    while (x != 0)
    {
        if (x % 10 < min) min = x % 10;
        x /= 10;
    }
    return min;
}

static uint Valoare(uint x, uint min)
{
    uint y = 0, p = 1;
    while (x != 0)
    {
        if (x % 10 != min)
        {
            y = y + (x % 10) * p;
            p *= 10;
        }
        x /= 10;
    }
    return y;
}
}

```



C:\Windows\system32\cmd.exe

```

Dati un numar intreg pozitiv : 176176
Eliminand cifra minima 1 din 176176 obtinem 7676
Press any key to continue . . .

```