

\*\*\*\*\*

# Tema 4

\*\*\*\*\*

- Primena operatora nad bitovima u zadacima
- Binarni zapis negativnih brojeva
- Algoritmi u C#

## PRIMER - Koju vrednost ima bit na poziciji 4 broja 155?

### POSTUPAK:

```
// a>>(poz-1)
// a & 1
```

```
using System;
```

```
namespace Bitovi3
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            int x = 155;
```

```
            int pozicija = 7;
```

```
            int maska = 1 << (pozicija - 1);
```

```
            // izracunamo vrednost bita
```

```
            int rezultat = x & maska;
```

```
            // vratimo bit
```

```
            rezultat = rezultat >> (pozicija - 1);
```

```
            PrintInBinary(x);
```

```
            Console.WriteLine(rezultat);
```

```
        }
```

```
        static void PrintInBinary(int num)
```

```
        {
```

```
            byte[] bytes = BitConverter.GetBytes(num);
```

```
            for (int count = bytes.Length - 1; count >= 0; count--)
```

```
            {
```

```
                for (int position = 7; position >= 0; position--)
```

```
                    PrintBitAt(bytes[count], position);
```

```
                Console.Write(" ");
```

```
            }
```

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

```
        }
```

```
        static void PrintBitAt(byte b, int position)
```

```
        {
```

```
            if (position > 7)
```

```

        return;
    byte a = (byte)Math.Pow(2, position);
    if ((b & a) == a)
        Console.WriteLine("1");
    else
        Console.WriteLine("0");
    }
}
}

```

## OBJASNJENJE RESENJA:

NAPOMENA 1: Objasniti i na tabli uz ostale operatore nad bitovima

NAPOMENA 2: Funkcije za ispis bitova bi trebalo kopirati i u naredne fajlove.

**PRIMER 2 - Sigurno postavljanja bita na broj 1. Ako je dat broj 12 u binarnom obliku, promeniti vrednost bita na poziciji 5. Koji broj se dobije? 28**

```

        // 1 << (poz - 1)
        // 12 | 1

static void Main(string[] args)
{
    int x = 12;
    int pozicija = 5;

    int maska = 1 << (pozicija - 1);
    // izracunamo vrednost bita
    int rezultat = x | maska;

    PrintInBinary(x);
    Console.WriteLine(rezultat);
}

```

NAPOMENA 1: Svi bitovi, osim maskiranog, zadržaće svoje vrednosti, dok će maskirani bit posle izvršavanje operacije sigurno imati vrednost 1.

**PRIMER 3 - Isključiti bit sa 1 na 0 : za broj 33 isključiti bit na poziciji 6. Koji broj se dobija tada?**

```

        // m = 1 << (poz-1)
        // ~m
        // a & ~m

static void Main(string[] args)
{
    int x = 33;
    int pozicija = 6;
}

```

```

        int maska = 1 << (pozicija - 1);
        // ~m
        // a & ~m
        int rezultat = ~maska;

        rezultat = rezultat & x;

        PrintInBinary(x);
        Console.WriteLine(rezultat);
    }

```

**NAPOMENA 1:** Rezultat treba da bude broj 1.

#### **PRIMER 4 - invertovanje vrednosti bita na poz 5 broja 223.**

```

using System;
namespace bitovi1
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 223;
            int pozicija = 5;

            PrintInBinary(x);

            int maska = 1 << (pozicija - 1);
            // m = 1 << (poz - 1)
            // a ^ m

            int rezultat = x ^ maska;

            Console.WriteLine(rezultat);
        }
    }
}

```

**NAPOMENA 1:** Za svaki primer ispisati i na tabli kako to izgleda.

#### **PRIMER 5: Zapisi negativnih brojeva u racunarstvu**

##### **Nacin 1: Metoda znaka i apsolutne vrednosti**

Npr: -7:

->bit na levom kraju oznacava bit znaka: 0 je pozitivvan broj, 1 je negativan broj

->7 se zapise kao 0000 0111

-> -7 je onda 1000 0111

## Nacin 2: Metoda dvojnog komplementa

Npr: -7

-> bit na zadnjem kraju predstavlja neg vrednos stepena dvojke.

-> 1111 0000 :  $-128+64+32+16 = -128+112 = -16$

## Kako pravimo zapis za negativni broj?

Npr -13

- 1) Napisemo pozitivan > 13: 00001101
- 2) Invertujemo 11110010
- 3) Dodamo broj jedan 00000001 i dobijemo: 11110011
- 4) To je taj broj
- 5) Ako hocemo da proverimo:  $-128+64+32+16+2+1 = -13$

## PRIMER 7: Ispisivanje negativnog broja preko niza karaktera

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine(GetIntBinaryString(-13));
    }
    static string GetIntBinaryString(int n)
    {
        char[] b = new char[32];
        int pos = 31;
        int i = 0;

        while (i < 32)
        {
            if ((n & (1 << i)) != 0)
            {
                b[pos] = '1';
            }
            else
            {
                b[pos] = '0';
            }
            pos--;
            i++;
        }
        return new string(b);
    }
}
```