

1. Program kojim se određuje da li je uneti prirodan broj savršen. (Broj je savršen ako je jednak sumi svojih delioca isključujući njega samog, a svi delioci su $< \text{ili} = \text{njegovoj polovini}$)

Npr. broj 6, 28, 496 su savršeni brojevi $28=1+2+4+7+14$

```
private void btSavrsen_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt32(tbn.Text), s=0;
    for (int i = 1; i <= n / 2; i++)
        if (n % i == 0) s += i;
    if (n == s) lbRezultat.Text = "Broj " + n + " je savršen";
    else lbRezultat.Text = "Broj " + n + " nije savršen";
}
}
```



2. Kreirati aplikaciju kojom se za dati prirodan broj N provrava da li je prost broj.

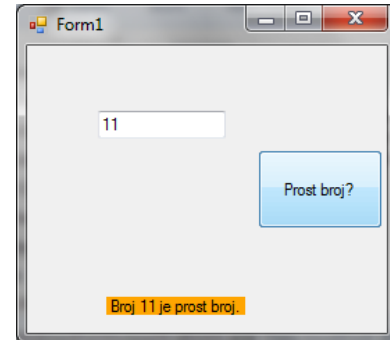
Napravljen je zaseban metod koji za argument ima prirodan broj N i proverava da li je broj prost. Vraća TRUE ako jeste, FALSE ako nije.

-svojstvo visible label1 lbRezultat postaviti na false

```
public partial class Form1 : Form
```

```
{
    public Form1()
    {
        InitializeComponent();
    }
    private bool prost(int n)
    {
        int i=2;
        if(n==1) return false; // 1 nije prost broj
        //ciklus se prekida kada je i delilac broja n ili kada se izadje iz intervala od 2 do Math.Sqrt(n)
        while(i<=Math.Sqrt(n) && n%i!=0)
            i++; // ako je i>Math.Sqrt(n) broj n nema delioce, znači da je prost, vraća se true
        return(i>Math.Sqrt(n));
    }

    private void btProst_Click(object sender, EventArgs e)
    {
        int n = Convert.ToInt32(tbn.Text);
        if (prost(n))
            lbRezultat.Text="Broj "+n+" je prost broj.";
        else lbRezultat.Text="Broj "+n+" nije prost broj.";
    }
}
}
```

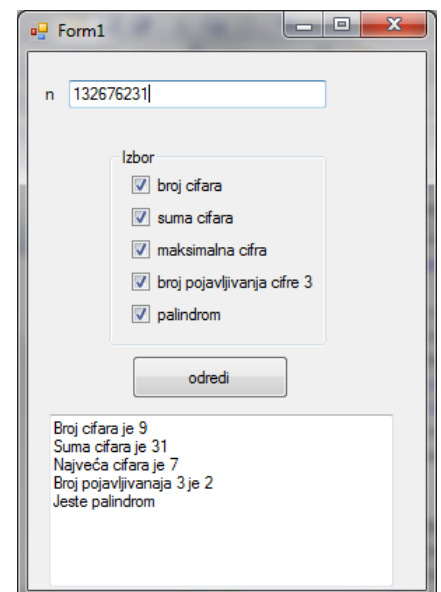


3. Napisati program koji za dati prirodan broj n određuje:

- kao broj cifara u dekadnom zapisu broja n;
- sumu cifara broja n;
- maksimalna cifra
- da li je broj n palindrom

Svojstvo TextBox2 multiline staviti na TRUE (pisanje u više redova)

```
private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt32(textBox1.Text);
    int c, s, max, bc, b3, x, inv;
    s = bc = max = b3 = x = 0;
    inv = n; //cuva vrednost broja n
    while (n > 0)
    {
        c = n % 10; //odredjuje poslednju cifru broja n
        n /= 10; //uklanja poslednju cifru iz broja n
        bc++; //uvecava broj cifara
        if (c > max) max = c; //odredjuje najveću cifru
        if (c == 3) b3++; // broj cifara jednakih 3
        s += c; // sabira cifre broja
        x = x * 10 + c; //formiranje inverznog broja
    }
}
```



```

        textBox2.Text = ""; // da ne ponavlja svaki put ispisane redove
        if (checkBox1.Checked) textBox2.Text += "Broj cifara je " + bc + "\r\n";
        if (checkBox2.Checked) textBox2.Text += "Suma cifara je " + s + "\r\n";
        if (checkBox3.Checked) textBox2.Text += "Najveća cifara je " + max + "\r\n";
        if (checkBox4.Checked) textBox2.Text += "Broj pojavljivanja 3 je " + b3 + "\r\n";
        if (checkBox5.Checked)
        {
            if (inv == x)
                textBox2.Text += "Jeste palindrom\r\n";
            else
                textBox2.Text += "Nije palindrom\r\n";
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        textBox2.Text = ""; // ako unosimo novi broj da se isprazni textbox2
    }

```

3. Napisati metode koji za 2 prirodna broja određuju:

- NZD najveći zajednički delilac;
- NZS najmani zajednički sadržalac;

NZD polazimo od manjeg od brojeva a i b i smanjujemo ga dok ne bude zajednički delilac ta dva broja

```

int NZD1(int a,int b)
{
    if (a==0)return b;
    if (b==0) return a; // ako je jedan od brojeva 0, onda je NZD drugi broj
    int x=a;
    if (x>b) x=b; // x dobija vrednost manjeg od 2 uneta broja
    while (a % x !=0 || b % x !=0)
        x--; // kada sa oba uslova netačna, tj. oba broja deljiva vraća vrednost x
    return x;
}

```

NZD se može odrediti kao proizvod zajedničkih prostih faktora a i b

```

int NZD2(int a,int b)
{
    if (a==0)return b;
    if (b==0) return a;
    int k=2, x=1;
    while (k<=a && k<=b)
        if (a%k == 0 && b%k == 0)
        {
            a=a/k;
            b=b/k;
            x=x*k;
        }
        else
            k++;
    return x;
}

```

```

int NZD3(int a,int b)
{
    if (a==0)return b;
    if (b==0) return a;
    while (a!=b)
        if (a>b)
            a=a-b;
        else
            b=b-a;
    return a;
}

```

Najefikasniji način za određivanje NZD je Euklidov algoritam

```
int nzd(int a, int b)
{
    if(a==0) return b;
    int p;
    while(b>0)
        {p=a;
         a=b;
         b=p%b;
        }
    return a;
}
```

http://sr.wikipedia.org/sr/Euklidov_algoritam

NZD dva broja je najveći broj koji istovremeno deli oba bez ostatka. Euklidov algoritam je zasnovan na principu da se najveći zajednički delilac dva broja ne menja ukoliko se manji broj oduzme od većeg, pa se zatim odredi NZD novodobijenog broja i manjeg od prethodna dva. Na primer, 21 je NZD za 252 i 105 ($252 = 21 \times 12$; $105 = 21 \times 5$); pošto je $252 - 105 = 147$, NZD za 147 i 105 je takođe 21. Kako je veći od dva polazna broja na ovaj način smanjen, ponavljanjem postupka dobijaće se sve manji brojevi, dok se jedan od njih ne svede na nulu. U tom trenutku, drugi broj je jednak najvećem zajedničkom deliocu dva polazna broja. Ukoliko se obrne redosled koraka u Euklidovom algoritmu, NZD se može izraziti kao zbir dva polazna broja od kojih je svaki pomnožen nekim celim brojem, u prethodnom primeru je $21 = 5 \times 105 + (-2) \times 252$. Ova važna osobina je poznata kao Bezuov identitet.

NZS dva broja najmanji od brojeva koji je deljiv bez ostatka sa oba broja Tri različita algoritma za određivanje NZS

```
int NZS1(int a, int b)
{
    return a*b / nzd(a,b); // uz pomoć metode NZD
}
```

ili polazimo od većeg od tih brojeva i uvećavamo ga za vrednost većeg broja dok ne bude NZS

```
int NZS2(int a, int b)
{
    int x;
    if (b>a)
        {
            x=b;
            b=a;
            a=x;
        }
    x=a;
    while (x%b !=0)
        x+=a;
    return x;
}
```

ili kao proizvod svih prostih faktora oba broj. Npr. NZS 24 i 16 je 48, dobija se kao proizvod $2*2*2*2*3$ (četiri dvojke su prosti faktori broja 16, a tri dvojke i trojka broja 24)

```
int NZS3(int a, int b)
{
    int k=2, x=1;
    while(k<=a || k<=b)
        if(a%k == 0 || b%k == 0)
            {
                if(a%k==0) a=a/k;
                if(b%k==0) b=b/k;
                x=x*k;
            }
        else
            k++;
    return x;
}
```