

01001111 01110011 01101110 01101111
01110110 01101001 00100000 01110000
01110010 01101111 01100111 01110010
01100001 01101101 01101001 01110010
01100001 01101110 01101010 01100001

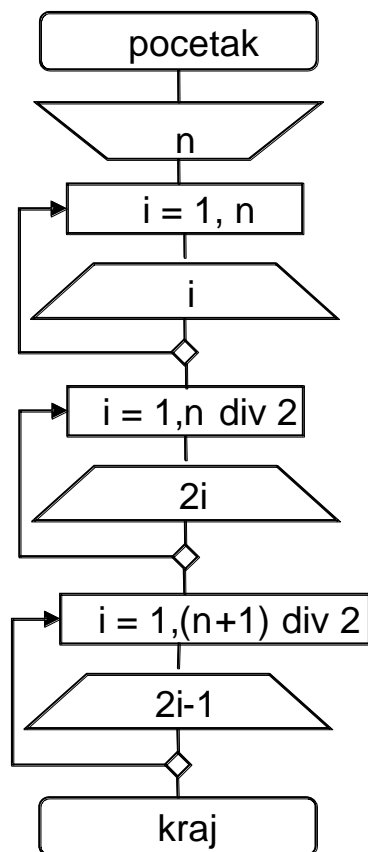


OSNOVI PROGRAMIRANJA

VEŽBE 5

Marina Svičević, Đorđe Nedić, Mladen Marić,
Danica Prodanović, Jovan Janićijević

- Napisati algoritam i program kojim se za uneti prirodan broj n ($n \geq 1$) prvo štampa sve prirodne brojeve od 1 do n , pa onda sve parne brojeve od 1 do n i na kraju sve neparne brojeve od 1 do n .



```

#include <stdio.h>

main()
{
    int i,n;

    scanf("%d",&n);

    printf("Svi brojevi od 1 do %d su : \n",n);
    for (i=1;i<=n;i++)
        printf("%3d ",i);

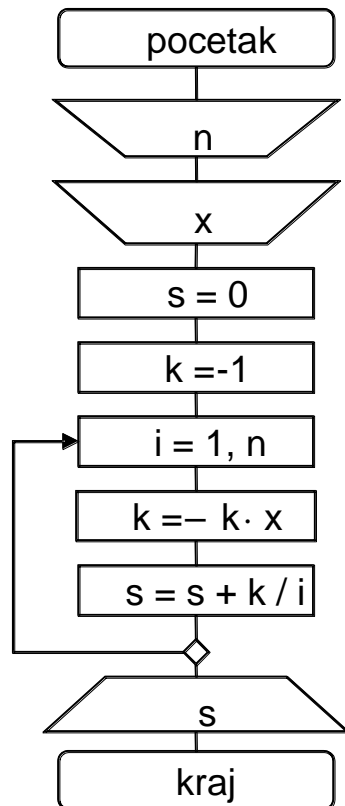
    printf("\n Svi parni od 1 do %d su : \n",n);
    for (i=1;i<=n/2;i++)
        printf("%3d ",2*i);

    printf("\n Svi neparni od 1 do %d su : \n",n);
    for (i=1;i<=(n+1)/2;i++)
        printf("%3d ",2*i-1);

}
  
```

- Napisati algoritam i program kojim se za uneti prirodan broj n ($n \geq 1$) i realan broj x , izračunava broj S na sledeći način:

$$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n-1} \frac{x^n}{n}$$



```

#include <stdio.h>
main()
{
    int i,n;
    float x,s,k;

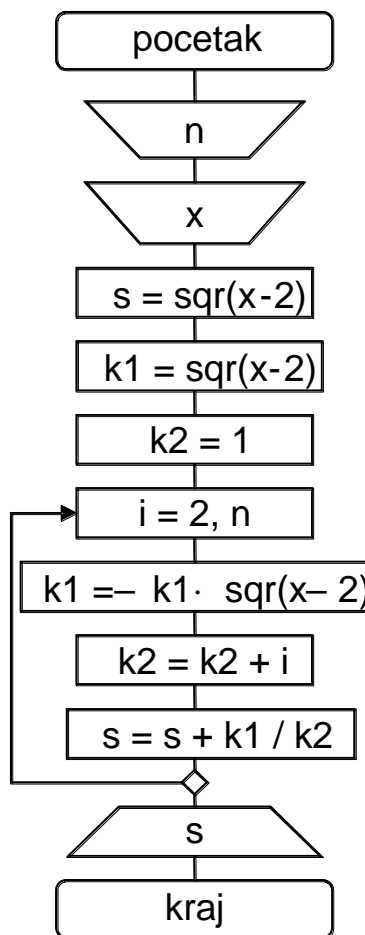
    scanf("%d",&n);
    scanf("%f",&x);

    s=0;
    k=-1;
    for(i=1;i<=n;i++)
    {
        k = -k*x;
        s += k/i;
    }
    printf("%10.6f\n",s);
}
  
```

- Napisati algoritam i program kojim se za uneti ceo broj n i broj x izračunava broj S na sledeći način:

$$S = \sum_{k=1}^n (-1)^{k-1} \frac{(x-2)^{2k}}{1+2+\dots+k}$$

$$S = \sum_{k=1}^n (-1)^{k-1} \frac{(x-2)^{2k}}{1+2+\dots+k}$$



```

#include <stdio.h>
#include <math.h>
main()
{
    int i,n;
    float x,s,k1,k2;

    scanf("%d",&n);
    scanf("%f",&x);

    s = pow(x-2,2);
    k1 = pow(x-2,2);
    k2 = 1;
    for(i=2;i<=n;i++)
    {
        k1 = -k1*pow(x-2,2);
        k2=k2+i;
        s += k1/k2;
    }

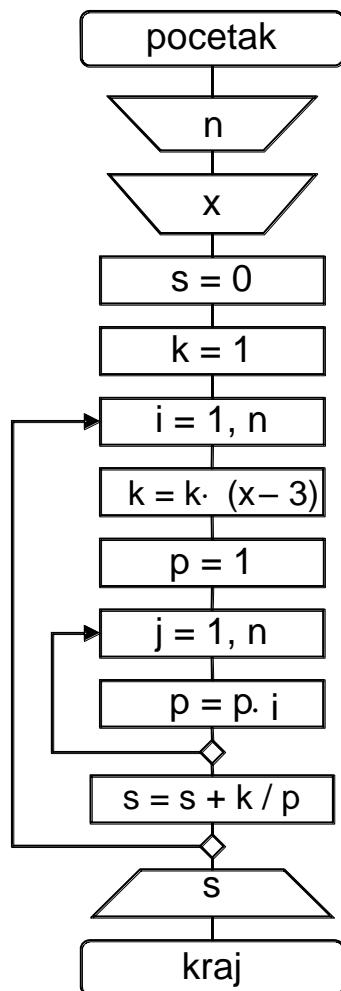
    printf("%10.6f\n",s);
}

```

- Napisati algoritam i program kojim se za uneti ceo broj n i broj x izračunava broj S na sledeći način:

$$S = \sum_{k=1}^n \frac{(x-3)^k}{k^n}$$

$$S = \sum_{k=1}^n \frac{(x-3)^k}{k^n}$$

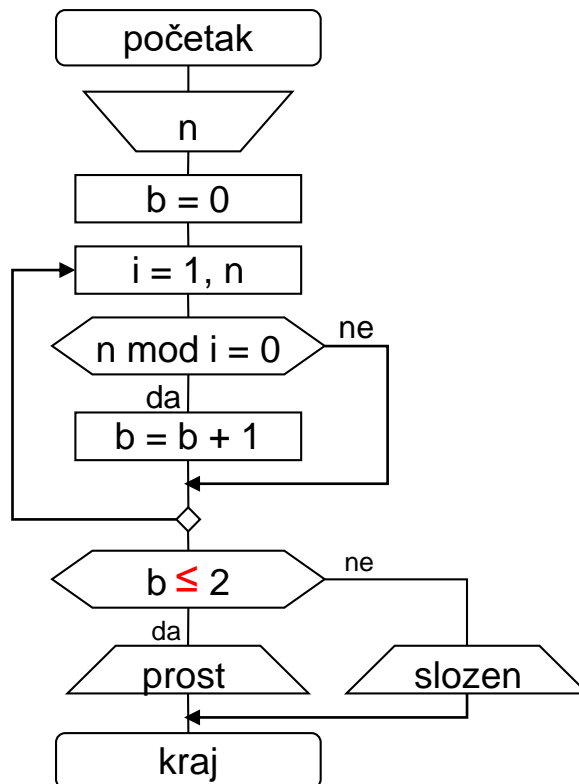


```

#include <stdio.h>
#include <math.h>
main()
{
    int i,j,n;
    float x,s,k;
    long p;

    scanf("%d",&n);
    scanf("%f",&x);
    s = 0;
    k=1;
    for(i=1;i<=n;i++)
    {
        k = k*(x-3);
        p=1;
        // Moze i for(j=1;j<=n;j++) p=p*i;
        for (j=0;j<n;j++) p *= i;
        s += k/p;
    }
    printf("%10.6f\n",s);
}
  
```


- Napisati algoritam i program kojim se za uneti ceo broj n ispituje da li je broj prost.

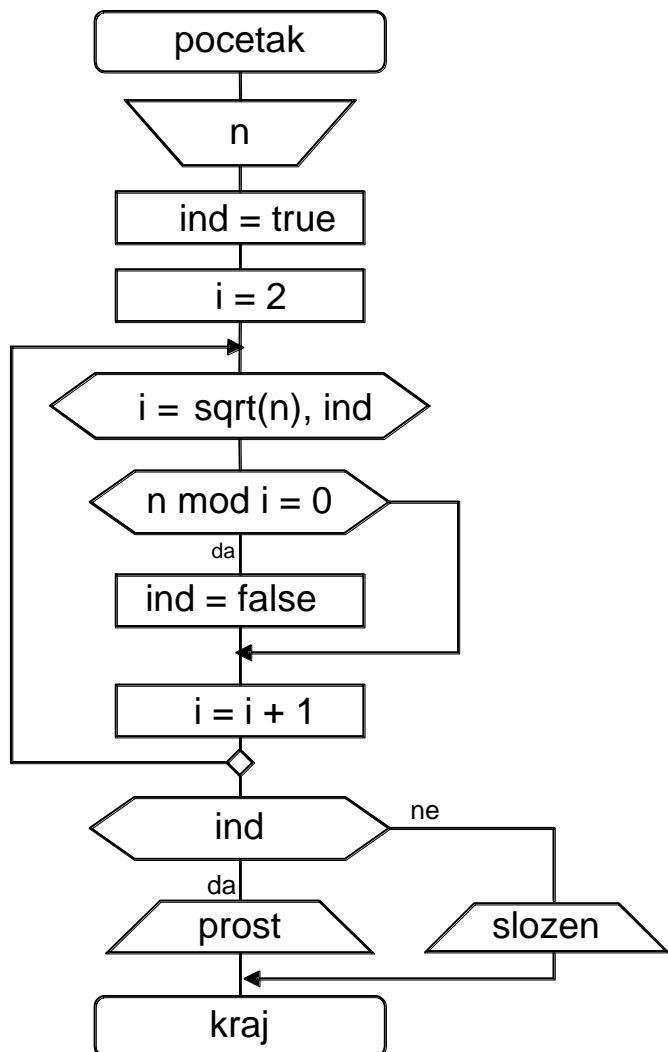


```

#include<stdio.h>

main()
{
    int n,i,b;

    scanf("%d",&n);
    b=0;
    for (i=1;i<=n;i++)
    {
        if (n % i == 0) b++;
    }
    if(b<=2) printf("Broj je prost \n");
    else printf("Broj je slozen \n");
}
  
```



```

#include <stdio.h>
#include <math.h>
main()
{
    int n,i,ind;

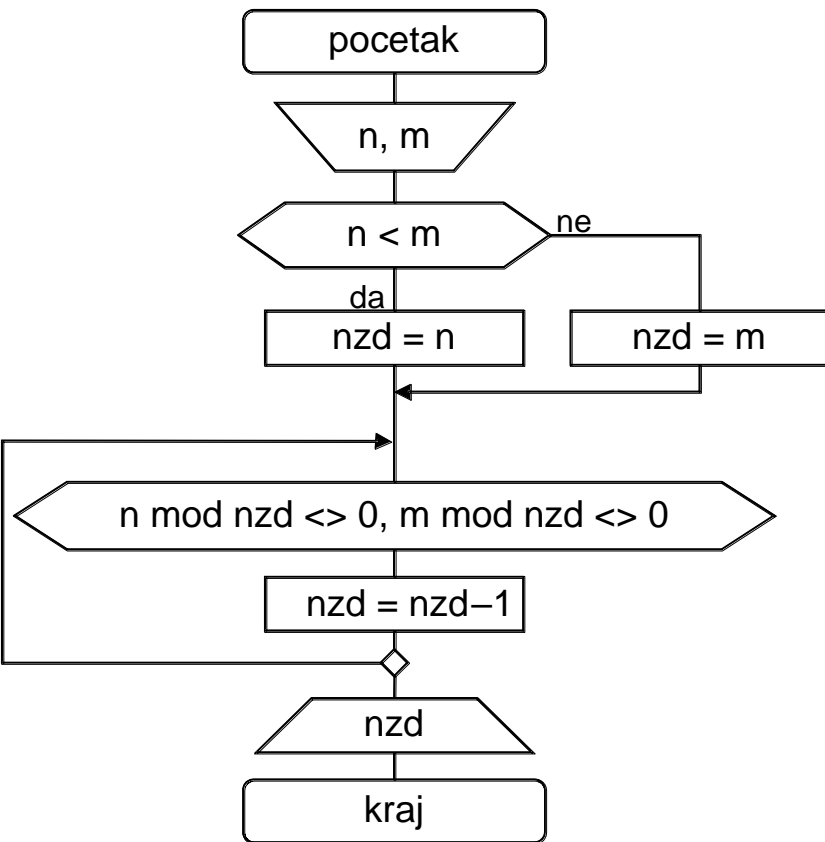
    scanf("%d",&n);
    i=2;

    while ((i <= sqrt(n)) && ind)
    {
        if (n % i == 0) ind=0;
        i+=1; // moze i i++;
    }

    if (ind) printf("Broj je prost \n");
    else printf("Broj je slozen \n");
}

```

- Napisati algoritam i program koji za uneta dva prirodna broja **m** i **n** odredjuje njihov NZD.



```

#include <stdio.h>

main()
{
    int n,m,nzd;

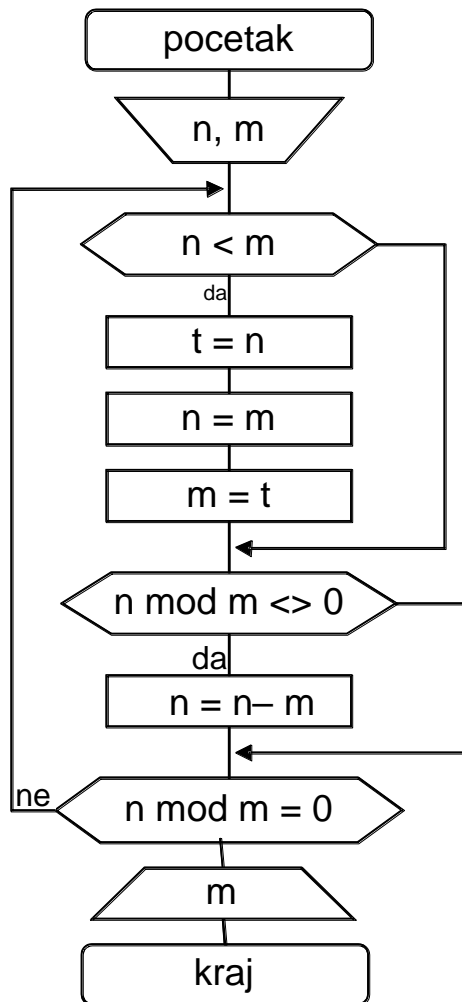
    scanf("%d%d", &n, &m);

    if (n<m) nzd=n;
    else nzd=m;

    while((m % nzd !=0) || (n % nzd !=0)) nzd--;

    printf("%d\n",nzd);

}
  
```



```

#include <stdio.h>

main()
{
    int n,m,t;

    scanf("%d%d",&n,&m);
    do
    {
        if (n<m)
        {
            t=n;
            n=m;
            m=t;
        }

        if (n % m != 0) n = n-m;
    }
    while (n % m != 0);

    printf("%d\n",m);
}
  
```