

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



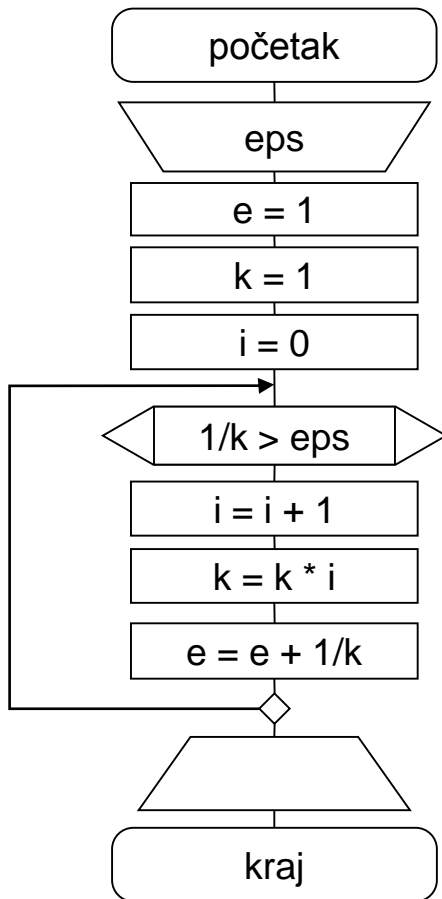
OSNOVI PROGRAMIRANJA

VEŽBE 6

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

- Napisati algoritam i program koji za zadatu tačnost eps izračunava Ojlerov broj e

$$e = \sum_{i=0}^{\infty} \frac{1}{i!}$$



```

#include <stdio.h>
main()
{
    int i;
    float k,e,eps;

    scanf("%f",&eps);
    e=1;
    k=1;
    i=0;
    while (1/k>eps)
    {
        i++;
        k *=i;
        e +=1/k;
    }
    printf("%6.2f\n",e);
}
  
```

```

do
{
    i++;
    k *=i;
    e +=1/k;
}
while (1/k>eps);
  
```

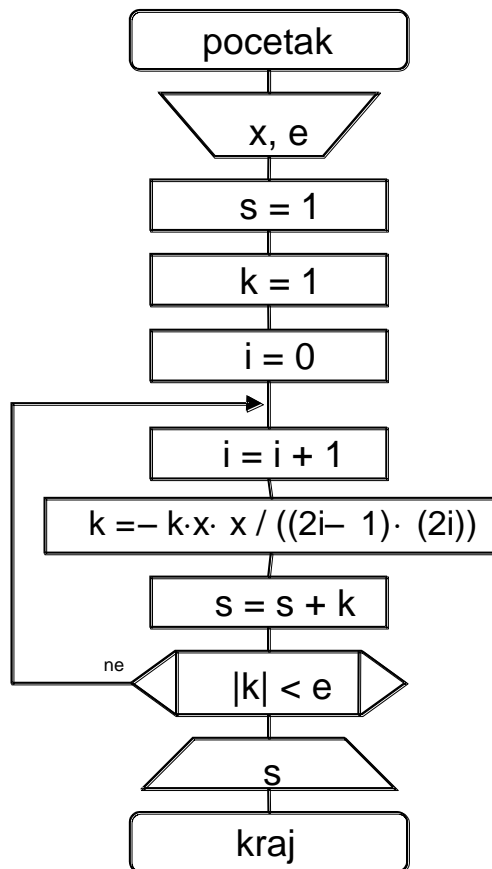
- Napisati algoritam i program kojim se za uneti realan broj x približno izračunava $\cos x$, za zadatu tačnost E , koristeći formulu

$$\cos(x) = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{(2i)!}$$

$$a_{i-1} = (-1)^{i-1} \frac{x^{2i-2}}{(2i-2)!} \quad a_i = (-1)^i \frac{x^{2i}}{(2i)!} \Rightarrow \frac{a_i}{a_{i-1}} = \frac{(-1)^i \frac{x^{2i}}{(2i)!}}{(-1)^{i-1} \frac{x^{2i-2}}{(2i-2)!}}$$

$$\Rightarrow \frac{a_i}{a_{i-1}} = -\frac{x^2}{(2i-1)(2i)}$$

$$\frac{a_i}{a_{i-1}} = -\frac{x^2}{(2i-1)(2i)}$$



```

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

    scanf("%f%f", &x, &eps);
    s=1.0;
    k=1.0;
    i=0;
    do
    {
        i++;
        k = -k*x*x/((2*i-1)*(2*i));
        s=s+k;
    }
    while (fabs(k)>=eps);

    printf("%10.6f", s);

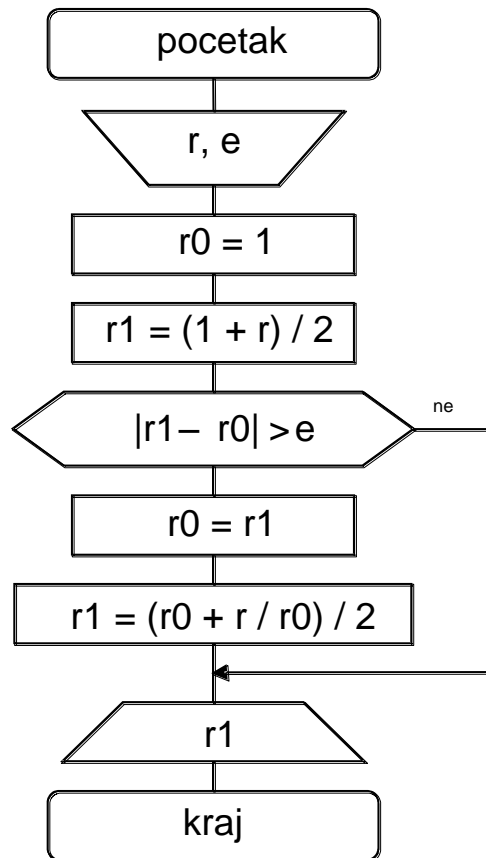
}
  
```

- Napisati algoritam i program kojim se za uneti realan broj R približno izračunava \sqrt{R} , za zadatu tačnost E , koristeći formulu

$$R_n = \frac{R_{n-1} + \frac{R}{R_{n-1}}}{2}; R_0 = 1$$

$$R_0 = 1, R_1 = \frac{1+R}{2}, R_2 = \frac{\frac{1+R}{2} + \frac{R}{\frac{1+R}{2}}}{2}$$

$$R_n = \frac{R_{n-1} + \frac{R}{R_{n-1}}}{2}; R_0 = 1$$



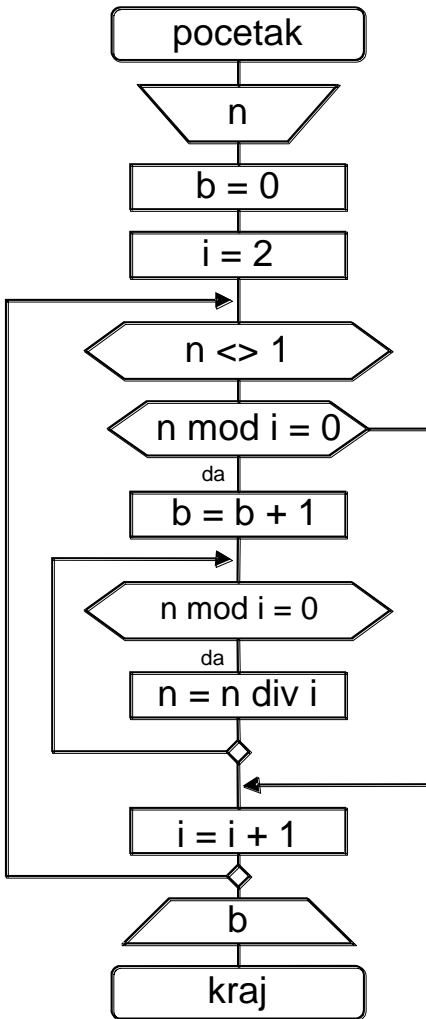
```

#include <stdio.h>
#include <math.h>
main() {

    double e, r0, r1, r;

    scanf("%lf%lf", &r, &e);
    r0=1;
    r1=(r+1)/2;
    while(fabs(r1-r0)>e)
    {
        r0=r1;
        r1=(r0+r/r0)/2;
    }
    printf("%10.6lf\n", r1);
}
  
```

- Napisati algoritam i program koji za uneti prirodan broj x određuje broj prostih delilaca.



```

#include <stdio.h>

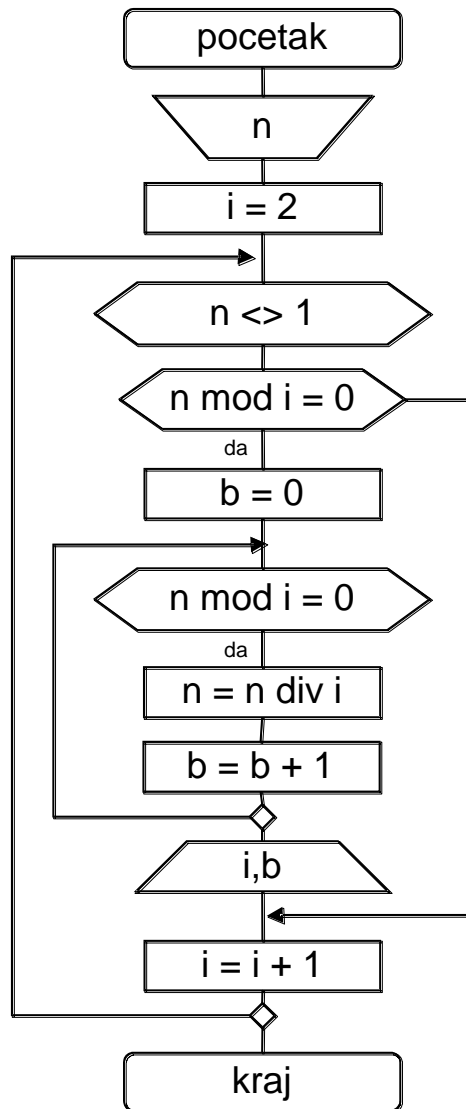
main()
{
    int n,i,b;

    scanf("%d",&n);
    i=2;
    b=0;

    while (n!=1)
    {
        if (n%i ==0)
        {
            b++;
            while (n % i ==0) n/=i; //ili n=n/i;
        }
        i++;
    }

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


- Napisati algoritam i program koji za uneti prirodan broj x radi njegovu faktORIZACIJU.



```

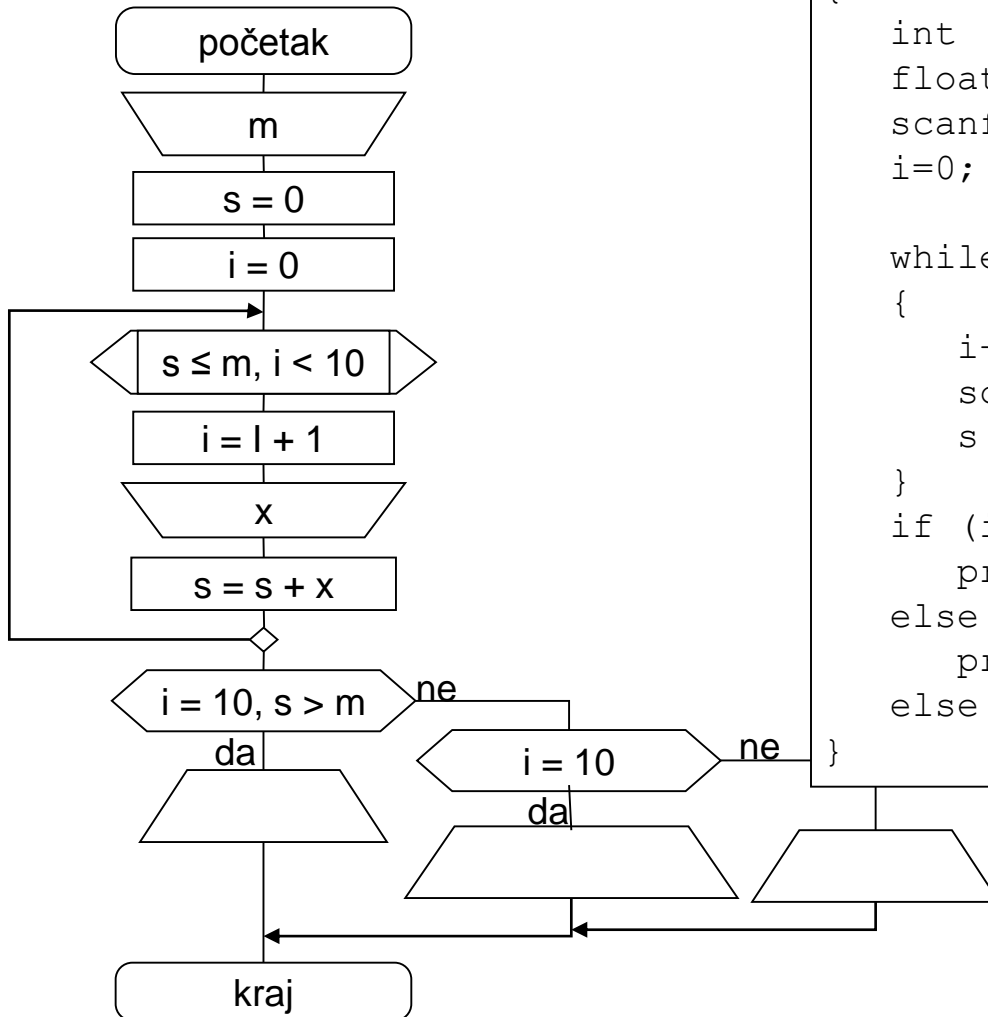
#include <stdio.h>

main() {

    int n,i,b;

    scanf("%d",&n);
    i=2;
    while (n!=1)
    {
        if ( n % i ==0)
        {
            b = 0;
            while (n % i ==0)
            {
                n/=i;
                b=b+1;
            }
            printf("%d na %d \n",i,b);
        }
        i++;
    }
}
  
```

- Napisati algoritam i program u kome se najpre unosi realan broj m , a zatim se unose realni brojevi sve dok zbir unetih brojeva ne postane veći od m ili dok se ne unese 10 brojeva. Na izlazu ispisati koji je kriterijum za izlaz iz petlje ispunjen.



```

#include <stdio.h>
main()
{
    int i;
    float m,x,s;
    scanf("%f",&m);
    i=0; s=0;

    while (s<=m && i<10)
    {
        i++;
        scanf("%f",&x);
        s +=x;
    }
    if (i==10 && s>m)
        printf("Ispunjena su oba uslova");
    else if (i==10 )
        printf("Uneto je 10 brojeva");
    else printf("Suma je veca od %6.2f",m);
}
  
```

- Napisati algoritam i program kojim se unosi ceo broj n i n realnih brojeva i određuje najveći uneti broj.
- Napisati algoritam i program kojim se za date prirodne brojeve m i n određuje suma m poslednjih cifara broja n .
- Napisati algoritam i program koji za uneta dva prirodna broja m i n određuje njihov NZS.