

## Peti zadatak

type `podaci.m`

```
X=[-5, -4, -2, -1, 2, 3, 5, 6, 7, 8, 9, 10];  
  
f=@(x)(x+1)/3;
```

type `tablica.m`

```
function [X1 Y1]=tablica()  
podaci;  
X1=X(find(mod(X,2)==0));% prvo nadjemo indekse sa parnim  
                           %elementima i samo njih smestamo u vektor X1  
Y1=round(f(X1),3);  
  
% funkcija round(a) zaokružuje broj a na najblizi ceo broj          round(3.2)=3  
%funkcija ceil(a) zaokružuje broj a na najblizu vecu vrednost      ceil(3.2)=4  
% funkcija floor(a) zaokružuje broj a na najblizu manju vrednost   floor(3.8)=3
```

type `inverz`

```
function x=inverz(y)  
[X,Y]=tablica();  
disp([X,Y])  
[X1, I]=sort(Y);%vektor X1 je vektor Y sortira rastuce  
                %vektor je vektor sortiranja tj. vektor takav da je  
                % Y(i)=X1,...recimo sadrzi indeks vektora Y ciji se element  
                % pojavljuje prvi u vektoru X1, pa indeks vektora Y ciji se  
                % element pojavljuje kao drugi element u vektoru X, itd do  
                % indeksa elementa Y koji se poslednji pojavljuje u vektoru X1.  
Y1=X(I); % nova tablica je X1,Y1. elementi vektora Y1 su X(I).  
  
% [X1 I]= sort(Y)  
% X1=vektor Y sortiran rastuce  
% I= vektor t.d. Y(I)=X1,  
  
disp([X1 Y1])  
  
x=Lagr(y,X1,Y1);  
  
function y=Lagr(x,X,Y) % funkcija vraca vrednost Lagranzovog interpolacionog polinoma  
                        % u tacki x na osnovu tablice [X,Y].  
n=length(X);  
y=0;  
for i=1:n  
    p=1;  
    for j=1:n  
        if i~=j  
            p=p*(x-X(j))/(X(i)-X(j));  
        end  
    end  
    y=y+p*Y(i);  
end
```

## Sesti zadatak

type `funkcija.m`

```
f = @(x) x.^2+x./2-exp(x)./4;
```

```
% poziv:
% x=nula(1,4.5,10)
% x = 4.4997
```

```
%provera: ugradjena funkcija fzero(f,a)- nalazi nulu (anonimne) funkcije f u blizini
%tacke
```

```
type tablica1.m
```

```
function [X,Y]=tablica1(a,b,n)
funkcija;
X=linspace(a,b,n);
Y=f(X);
```

```
type promenaZnaka.m
```

```
function [c,d]=promenaZnaka(a,b,n)
[X,Y]=tablica1(a,b,n);
i=1;
while (i<n && Y(i)*Y(i+1)>0)
    i=i+1;
end
c=X(i);
d=X(i+1);
```

```
type nula.m
```

```
function x=nula(a,b,n)
[c,d]=promenaZnaka(a,b,n);
funkcija;
% Nakon tabeliranja na segmentu [c,d] nula ce biti na kraju tablice

[X1,Y1]=tablica1(a,b,n);
i=find(X1==d);
X1=X1(i-3:i);
Y1=Y1(i-3:i);
m=length(X1);
kr=zeros(m,m-1);
for i=1:m-1
    kr(i,1)=Y1(i+1)-Y1(i);
end
for j=2:m
    for i=1:m-j
        kr(i,j)=kr(i+1,j-1)-kr(i,j-1);
    end
end
disp([X1' Y1' kr])
% ODGOVARAJUCA ITERATIVNA FORMULA
% q =( y-Y(n)-q*(q+1)*kRazlike(n-2,2)-
%      -q*(q+1)*(q+2)*kRazlike(n-3,3))/kRazlike(n-1,1)

q0=0;
q1=(-Y1(m)-q0*(q0+1)*kr(m-2,2)/2-q0*(q0+1)*(q0+2)*kr(m-3,3)/6)/kr(m-1,1);
Q=[q0 q1];
while(abs(q0-q1)>=1e-4)
    q0=q1;
    q1=(-Y1(m)-q0*(q0+1)*kr(m-2,2)/2-q0*(q0+1)*(q0+2)*kr(m-3,3)/6)/kr(m-1,1);
    Q=[Q q1]; % U Q upisujemo svako q_i iz iterativnog procesa
end
disp('Nula funkcije je: ');
x = X1(end)+Q(end)*(X1(2)-X1(1));
```

```
disp(x);  
disp('iterativni niz q je: ');  
disp(Q');
```