

% helpUNM ce vam dati ideje kako da resite zadatke sa praktikuma.  
% ovo su samo ideje, naravno da to nije jedino resenje..

% TABLICA GENERISE DVA NIZA

% tablica.m

```
X = [1 2 3 4 5 6];  
Y = [2 5 10 17 26 37]; % na primer Y = X^2+1
```

% FUNKCIJA ZADATA EKSPLICITNO

% funkcija.m

```
f = inline('x.^2+1');
```

% ili

```
function y = funkcija(x)
```

```
y = x^2+1;
```

```
end
```

% tablica.m

```
% tabelira eksplicitno zadatu funkciju
```

```
% k cvorova, rastojanje h=1/n
```

```
function [X Y] = tablica(a,k,n)
```

```
funkcija;
```

```
X = zeros(1,k);
```

```
Y = zeros(1,k);
```

```
h = 1/n;
```

```
for i=1:k
```

```
    X(i) = a + (i-1)*h;
```

```
    % ako je funkcija definisana na I način
```

```
    Y(i) = feval(f,X(i));
```

```
    % ako je funkcija definisana na II način
```

```
    Y(i) = funkcija(X(i));
```

```
end;
```

% KONACNE RAZLIKE

```
kRazlike = zeros(n-1,n); % tablica; n= length(X);
```

```
for i = 1:n-1
```

```
    kRazlike(i,1) = Y(i+1)-Y(i);
```

```
end
```

```
for j = 2:n-1
```

```
    for i = 1:n-j
```

```
        kRazlike(i,j) = kRazlike(i+1,j-1) - kRazlike(i,j-1);
```

```
    end;
```

```
end;
```

% PODELJENE RAZLIKE

```
pRazlike = zeros(n,n-1); % tablica; n= length(X);
```

```
for i=1:n-1
```

```
    pRazlike(i,1) = (Y(i+1)-Y(i))/(X(i+1)-X(i));
```

```
end;
```

```
for j = 2:n-1 % red podeljenih razlika
```

```

    for i=1:n-j
        pRazlike(i,j) = (pRazlike(i+1,j-1)-pRazlike(i,j-1))/(X(i+j)-X(i));
    end;
end;

disp('Stampamo tablicu po kolonama : ');
disp([X' Y' pRazlike]);

```

## % INTERPOLACIONI POLINOMI

### % lagr.m

```

function lagr = lagr(x)

    tablica;
    n = length(X);
    L = 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;
        L = L+p*Y(i);
    end;
    lagr = L;

```

### % njutn1.m

```

function nj1 = njutn1(x)
    .....
    y = Y(1);
    h = X(2)-X(1);
    q = (x-X(1))/h;
    q1 = q;
    fakt = 1;
    for j = 1:n-2
        y = y + kRazlike(1,j)*q1/fakt;
        fakt = fakt*(j+1);
        q1 = q1*(q-j);
    end;

    nj1 = y;

end

```

```
% njutn2.m
```

```
function nj2= njutn2(x)
....
y = Y(end);
h = X(2)-X(1);
q = (x-X(end))/h;
fakt = 1;
for j = 1:n-2
    y = y + kRazlike(j,n-j)*q/fakt;
    fakt = fakt*(j+1);
    q = q*(q+j);
end;

nj2 = y;
```

```
% njutn3.m      Njutn sa podeljenim razlikama
```

```
function nj3 = njutn3(x)
....
y = Y(1);
p = 1;
for i = 1:n-1
    p = p*(x-X(i));
    y = y + pRazlike(1,i)*p;
end;
nj3 = y;
```

```
% inverz.m
```

```
% eventualni problem je kod sortiranja tablice..
% ako nije dato da je tablica po Y monotono rastuca
% onda je sortiramo na sl nacin:
tablica;
n = length(X);
% pravimo novu tablicu, Y i X menjaju mesta
nTablica = zeros(2,n);
nTablica(1, :) = Y;
nTablica(2, :) = X;

% sortiramo novu tablicu po 1.vrsti (Y vrsta)
% funkcija sortrows sortira kolone, zato transponujemo matricu
% i sortiramo je po kolonama (nase vrste)
% ' koristimo za transponovanje
% na kraju opet transponujemo da bi se vratili na stari oblik..
[Y1 X1] = (sortrows(nTablica', 1))';

disp([Y1 X1]); % stampamo sredjenu tablicu

.... % .....
```

```
% INTEGRACIJA
```

```
% trapez.m
```

```
function trapez = trapez()  
    % videti simpson.m pa analogno napisati trapez.m
```

```
% simpson.m
```

```
function I = simpson(f,a,b)  
    % funkcija f na intervalu [a,b]  
    % napravite pomocni vektor X1 sa n ekv. cvorova  
    h = (b-a)/(n-1);  
  
    I1 = feval(f,X1(1))+feval(f,X1(end)); % Y(0)+Y(2n)  
  
    I4 = 0; % Y(1)+Y(3)+Y(5)+Y(7)+...+Y(2n-1)  
    for i = 2:2:2*n-1  
        I4 = I4 + feval(f,X1(i));  
    end;  
  
    I2 = 0; % Y(2)+Y(4)+Y(6)+Y(8)+...+Y(2n-2)  
    for i = 3:2:2*n-2  
        I2 = I2 + feval(f,X1(i));  
    end;  
  
    I = h/3*(I1+4*I4+2*I2);
```