#include <iostream>

#include <cmath>

using namespace std;

int main() {

cout << "Implementing Newton Rapshon Method……………………………………………………………….."<<endl;

cout<<endl<<"Enter the degree of the equation:";

int n;

cin>>n;

double coeff[100];

cout<<"Enter coefficients.......................";

for(int i=n;i>=0;i--)

{

cout<<endl<<"Enter Coefficient for term with degree "<<i<<" :";

cin>>coeff[i];

}

double s[]={0,0};

int flag=0;

double specific[2];

double i=0;

double j=0;

while(flag==0)

{

s[0]=0;

s[1]=0;

cout<<endl<<i<<" "<<j;

for(int t=0;n-t>=0;t++) {

s[0] =s[0]+ (pow(i, n - t) \* coeff[n - t]);

s[1] =s[1]+ (pow(j, n - t) \* coeff[n - t]);

}

cout<<endl<<i<<":"<<s[0];

cout<<endl<<j<<":"<<s[1];

if(s[0]\*s[1]<=0)

{

cout << endl << "Root exsistence between " << i << " and " << j;

specific[0] = s[0];

specific[1] = s[1];

flag = 1;

break;

}

i++;

j--;

}

cout<<endl<<"Now finding values..............";

double tempval1,tempval2,tempval3;

tempval3=specific[0];

cout<<endl<<"Let i=x0..............";

double x=i;

double xh;

double h=0.00001;

double derivativeval,xn1;

for(int i=0;i<5;i++)

{

xh=x+h;

for(int t=0;n-t>=0;t++) {

tempval1 =tempval1+ (pow(x, n - t) \* coeff[n - t]);

tempval2=tempval2+ (pow(xh, n - t) \* coeff[n - t]);

}

derivativeval=(tempval2-tempval1)/h;

xn1=x-(tempval1/derivativeval);

x=xn1;

}

cout<<endl<<"Final value of root is:";

cout<<x;

return 0;

}