**/\*WAP to find solution of Non-Linear equations by Bisection Method.\*/**

**#include <iostream>**

**#include <cmath>**

**#include <iomanip>**

**#include <cstring>**

**#define pi 3.14159265358979323846264338327950288419716939937510**

**#define e 2.71828182845904523536028747135266249775724709369995**

**using namespace std;**

**double y;**

**inline void maths\_function(double & x)**

**{**

**y = pow(x,7)-sin(x)-cos(x);//Put your function here**

**cout<<"\t\t"<<setw(9)<<setprecision(9)<<x;**

**cout<<"\t\t"<<setw(9)<<setprecision(9)<<y;**

**}**

**int main()**

**{**

**int k,error,counter;**

**double x1,x2,x3;**

**cout.precision(9);**

**while(1)**

**{**

**counter=0;**

**cout<<"\n\t\tBISECTION METHOD\n\n";**

**cout<<"Initial guess (a,b) where f(a) < 0 & f(b) > 0\n";**

**cout<<"\nEnter your initial guess (a) : ";**

**cin>>x1;**

**cout<<"\nEnter your second guess (b) : ";**

**cin>>x2;**

**cout<<"\nEnter tolerance (10^-k)";**

**cout<<"\nEnter k: ";**

**cin>>k;**

**cout<<"Approx. no of steps = "<<abs((k\*log(10)-log(abs(x1-x2)))/log(2))<<endl;**

**cout<<"\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";**

**error=10;**

**cout<<" SN\t\t a"<<"\t\t\t f(a)"<<"\t\t b"<<"\t\t\t f(b)"<<"\t\t\tc=(a+b)/2"<<"\t\t f(c)\n\n";**

**while (error>9)**

**{**

**x3=(x1+x2)/2;**

**cout<<" "<<++counter;**

**maths\_function(x1);**

**maths\_function(x2);**

**maths\_function(x3);**

**cout<<endl<<endl;**

**if (y<0)**

**{**

**x1=x3;**

**}**

**else x2=x3;**

**error = (int)trunc(abs(y\*pow(10,k)));//error is in order of 10^k**

**}**

**cout<<"\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";**

**}**

**return 0;**

**}**