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

**#include <iostream>**

**#include <cmath>**

**#include <iomanip>**

**#include <cstring>**

**#define pi 3.14159265358979323846264338327950288419716939937510**

**#define e 2.71828182845904523536028747135266249775724709369995**

**using namespace std;**

**double y,x;**

**inline void maths\_function(double & x\_1,double & x\_2)**

**{**

**double y\_1,y\_2;**

**y\_1 = cos(x\_1)+pow(e,-x\_1)-5;//Put your function here**

**y\_2 = cos(x\_2)+pow(e,-x\_2)-5;//Put your function here**

**x = (y\_2\*x\_1 - y\_1\*x\_2)/(y\_2-y\_1);**

**y = cos(x)+pow(e,-x)-5;//Put your function here**

**cout<<"\t\t"<<setw(9)<<setprecision(9)<<x\_1;**

**cout<<"\t\t"<<setw(9)<<setprecision(9)<<y\_1;**

**cout<<"\t\t"<<setw(9)<<setprecision(9)<<x\_2;**

**cout<<"\t\t"<<setw(9)<<setprecision(9)<<y\_2;**

**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;**

**cout.precision(9);**

**while(1)**

**{**

**counter=0;**

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

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

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

**cin>>k;**

**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\tc=(f(b).a+f(a).b)/(f(b)-f(a))"<<"\t f(c)\n\n";**

**while (error>9)**

**{**

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

**maths\_function(x1,x2);**

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

**x1=x2;**

**x2=x;**

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

**}**

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

**}**

**return 0;**

**}**