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

**#include <iostream>**

**#include <cmath>**

**#include <iomanip>**

**#include <cstring>**

**#define pi 3.14159265358979323846264338327950288419716939937510**

**#define e 2.71828182845904523536028747135266249775724709369995**

**using namespace std;**

**double x\_1,y\_1;**

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

**{**

**double y,dy;**

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

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

**x\_1 = x-y/dy;**

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

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

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

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

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

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

**}**

**int main()**

**{**

**int k,error,counter;**

**cout.precision(9);**

**while(1)**

**{**

**counter=0;**

**cout<<"\n\t\tNEWTON-RAPHSON METHOD\n\n";**

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

**cin>>x\_1;**

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

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

**cin>>k;**

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

**error=10;**

**cout<<" SN\t\t a(n)"<<"\t\t\t f(a(n))"<<"\t\t f'(a(n))"<<"\t\t a(n+1)"<<"\t\t\t f(a(n+1))\n\n";**

**while (error>9)**

**{**

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

**maths\_function(x\_1);**

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

**error = (int)trunc(abs(y\_1\*pow(10,k)));//error is in order of 10^k**

**}**

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

**}**

**return 0;**

**}**