

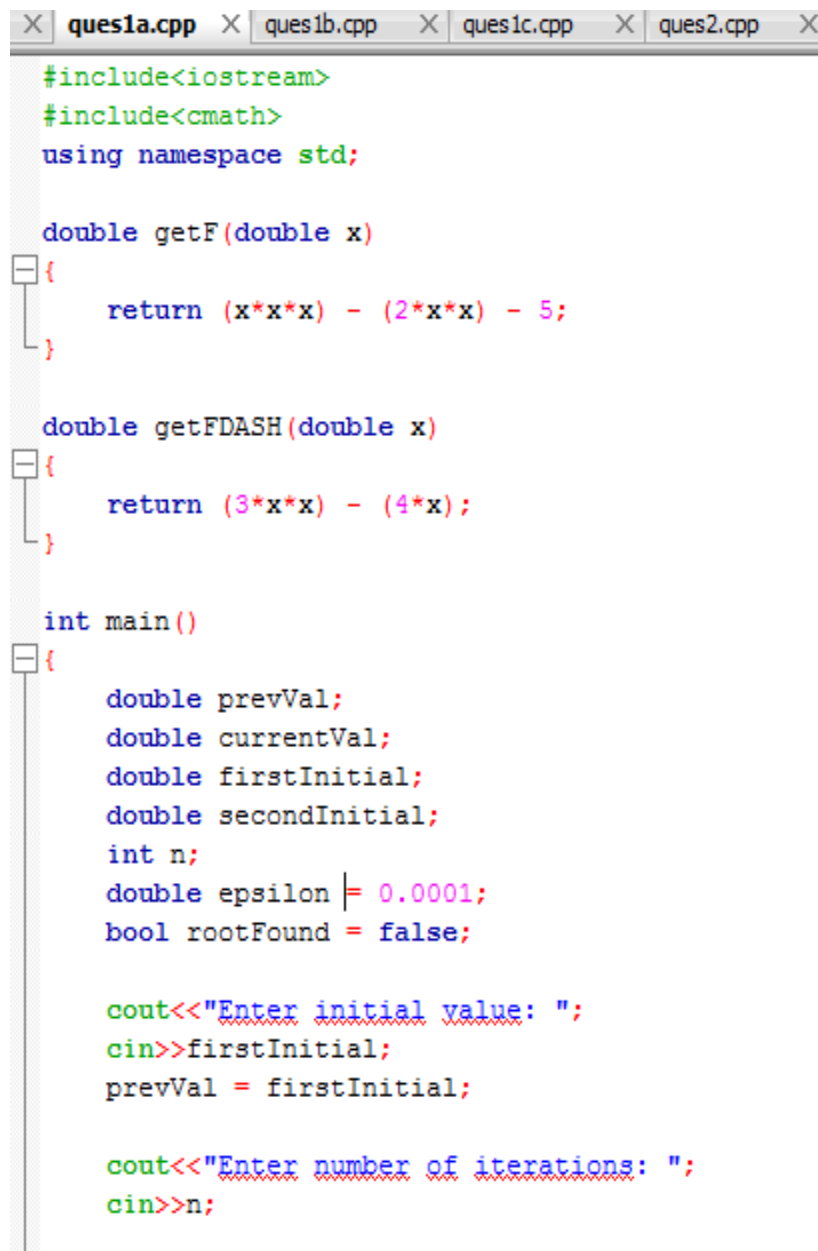
SCIENTIFIC COMPUTING

ASSIGNMENT 3

TUSHAR SIRCAR

130123038

1. SOURCE CODE:



```
ques1a.cpp  ques1b.cpp  ques1c.cpp  ques2.cpp
#include<iostream>
#include<cmath>
using namespace std;

double getF(double x)
{
    return (x*x*x) - (2*x*x) - 5;
}

double getFDASH(double x)
{
    return (3*x*x) - (4*x);
}

int main()
{
    double prevVal;
    double currentVal;
    double firstInitial;
    double secondInitial;
    int n;
    double epsilon = 0.0001;
    bool rootFound = false;

    cout<<"Enter initial value: ";
    cin>>firstInitial;
    prevVal = firstInitial;

    cout<<"Enter number of iterations: ";
    cin>>n;
```

```

cout<<"EVALUATING USING NEWTONS METHOD:\n";
for(int i=0; i<n; i++)
{
    currentVal = prevVal - (getF(prevVal)/getFDASH(prevVal));
    if(abs(currentVal - prevVal) < epsilon)
    {
        cout<<"Root found at: "<<currentVal;
        cout<<"\nNumber of iterations: "<<i+1;
        rootFound = true;
        break;
    }
    prevVal = currentVal;
    cout<<prevVal<<endl;
}

if(!rootFound)
    cout<<"Failed to converge!";

cout<<"\n\nEVALUATING USING SECANTS METHOD:\n";
rootFound = false;
cout<<"Enter second initial value: ";
cin>>secondInitial;
prevVal = firstInitial;
currentVal = secondInitial;
for(int i=0; i<n; i++)
{
    double temp = currentVal;
    currentVal = currentVal - ((getF(currentVal)*(currentVal-prevVal))/(getF(currentVal)-getF(prevVal)));
    prevVal = temp;
    if(abs(currentVal-prevVal) < epsilon)
    {
        cout<<"Root found at: "<<currentVal;
        cout<<"\nNumber of iterations: "<<i+1;
        rootFound = true;
        break;
    }
    cout<<prevVal<<endl;
}

if(!rootFound)
    cout<<"Failed to converge!";

return 0;

```

```

prevVal = temp;
if(abs(currentVal-prevVal) < epsilon)
{
    cout<<"Root found at: "<<currentVal;
    cout<<"\nNumber of iterations: "<<i+1;
    rootFound = true;
    break;
}
cout<<prevVal<<endl;
}

if(!rootFound)
    cout<<"Failed to converge!";

return 0;

```

(a) NEWTONS METHOD

Initial Value: 2.5

Root Found: 2.69065

Number Of Iterations: 4

SECANT METHOD

Initial Values: 2.5, 2.7

Root Found: 2.69065

Number Of Iterations: 3

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - [X]
Enter initial value: 2.5
Enter number of iterations: 200
EVALUATING USING NEWTONS METHOD:
2.71429
2.69095
2.69065
Root found at: 2.69065
Number of iterations: 4

EVALUATING USING SECANTS METHOD:
Enter second initial value: 2.7
Root found at: 2.69065
Number of iterations: 3
Process returned 0 (0x0)    execution time : 6.402 s
Press any key to continue.
```

(b) NEWTONS METHOD

Initial Value: 0

Root Found: -0.470064

Number Of Iterations: 6

SECANT METHOD

Initial Values: 0, -0.5

Root Found: -0.470064

Number Of Iterations: 5

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - [X]
Enter initial value: 0
Enter number of iterations: 200
EVALUATING USING NEWTONS METHOD:
-0.275112
-0.414193
-0.463047
-0.469928
-0.470064
Root found at: -0.470064
Number of iterations: 6
Value at this root: 7.37192e-015

EVALUATING USING SECANTS METHOD:
Enter second initial value: -0.5
-0.5
-0.489007
-0.468213
-0.470169
Root found at: -0.470064
Value at this root: 1.51093e-010
Number of iterations: 5
Process returned 0 (0x0)    execution time : 13.535 s
Press any key to continue.
```

NEWTONS METHOD

Initial Value: -100

Root Found: -2.64561

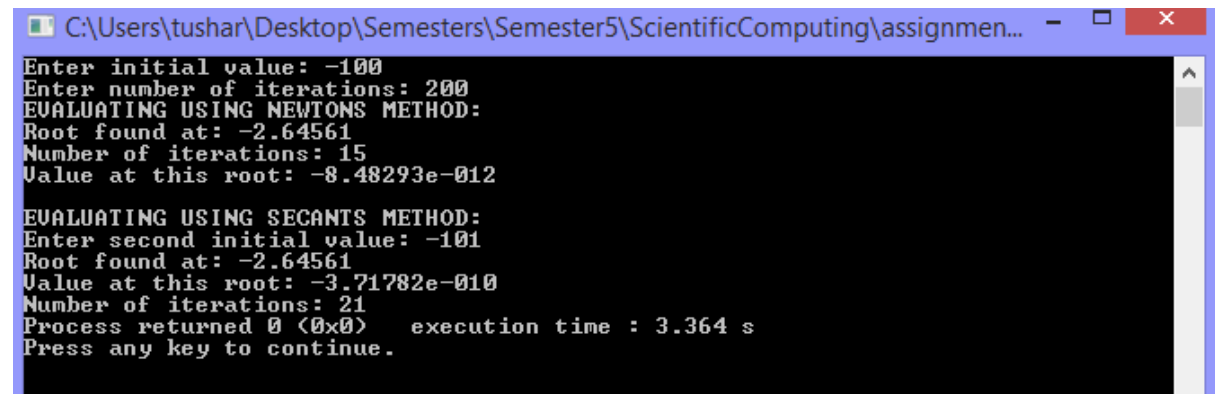
Number Of Iterations: 15

SECANT METHOD

Initial Values: -100, -101

Root Found: -2.64561

Number Of Iterations: 21



```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - [X]
Enter initial value: -100
Enter number of iterations: 200
EVALUATING USING NEWTONS METHOD:
Root found at: -2.64561
Number of iterations: 15
Value at this root: -8.48293e-012

EVALUATING USING SECANTS METHOD:
Enter second initial value: -101
Root found at: -2.64561
Value at this root: -3.71782e-010
Number of iterations: 21
Process returned 0 (0x0)    execution time : 3.364 s
Press any key to continue.
```

(c) NEWTONS METHOD

Initial Value: 1

Root Found: 1.49819

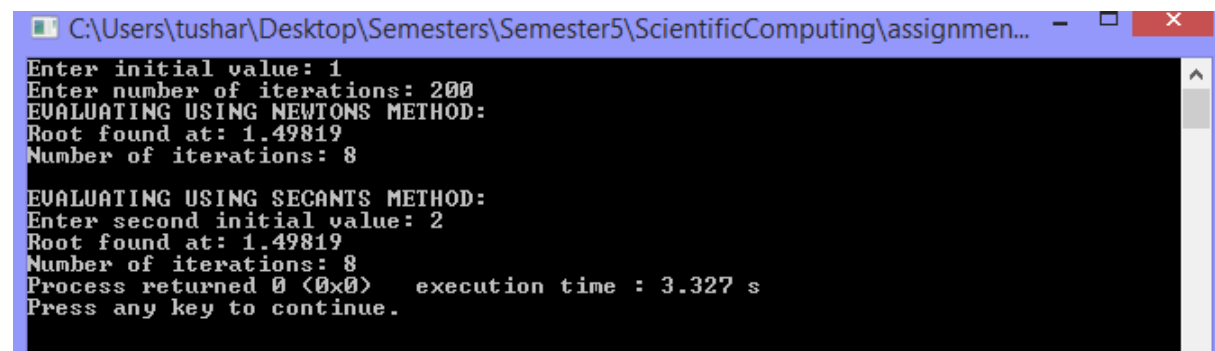
Number Of Iterations: 8

SECANT METHOD

Initial Values: 1, 2

Root Found: 1.49819

Number Of Iterations: 8



```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - [X]
Enter initial value: 1
Enter number of iterations: 200
EVALUATING USING NEWTONS METHOD:
Root found at: 1.49819
Number of iterations: 8

EVALUATING USING SECANTS METHOD:
Enter second initial value: 2
Root found at: 1.49819
Number of iterations: 8
Process returned 0 (0x0)    execution time : 3.327 s
Press any key to continue.
```

2. SOURCE CODE

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  double getF(double x)
6  {
7      return log((x*x)+1) - (exp(0.4*x)*cos(3.14*x));
8  }
9
10 int signum(double x)
11 {
12     if( x < 0)
13         return -1;
14     if( x > 0)
15         return 1;
16     return 0;
17 }
18
19 int main()
20 {
21     double low,high;
22     int n;
23     double epsilon = 0.000001;
24     bool rootFound = false;
25     double prevValue;
26
27     cout<<"Enter interval to find ONLY NEGATIVE ZERO: ";
28     cin>>low>>high;
29
30     cout<<"Enter number of iterations: ";
31     cin>>n;
32
33     for(int i=0; i<n; i++)
34     {
35         double mid = (low+high)/2;
36         if(signum(getF(mid)) == 0 || (i !=0 && (abs(mid-prevValue)/abs(mid))< epsilon) )
37         {
38             cout<<"Root found at: "<<mid;
39             cout<<"\nNumber of iterations: "<<i+1;
40             rootFound = true;
41             break;
42         }
43         if(signum(getF(low))*signum(getF(mid)) < 0)
44             high = mid;
45         else
46             low = mid;
47         prevValue = mid;
48     }
49     if(!rootFound)
50         cout<<"Failed to converge!";
51
52
53
54     return 0;
55 }
56
```

Negative Root: -0.434311

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - □ ×
Enter interval to find ONLY NEGATIVE ZERO: -1 0
Enter number of iterations: 200
Root found at: -0.434311
Number of iterations: 22
Process returned 0 (0x0)    execution time : 3.276 s
Press any key to continue.
```

First 4 positive roots:

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - □ ×
Enter value of n: 1
Enter number of iterations: 200
Root found at: 0.45085
Number of iterations: 22
Process returned 0 (0x0)    execution time : 4.159 s
Press any key to continue.
```

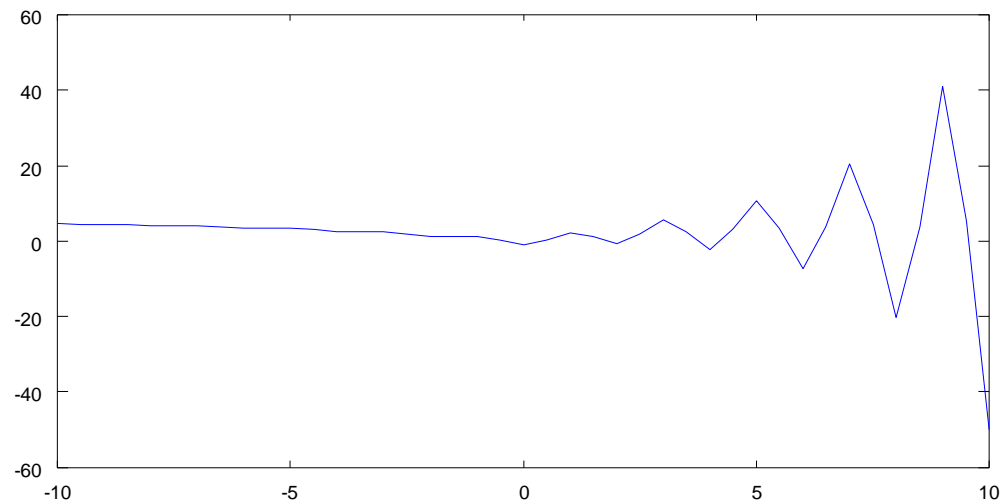
```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - □ ×
Enter value of n: 2
Enter number of iterations: 200
Root found at: 1.74569
Number of iterations: 20
Process returned 0 (0x0)    execution time : 1.120 s
Press any key to continue.
```

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - □ ×
Enter value of n: 3
Enter number of iterations: 200
Root found at: 2.23945
Number of iterations: 19
Process returned 0 (0x0)    execution time : 1.035 s
Press any key to continue.
```

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - □ ×
Enter value of n: 4
Enter number of iterations: 200
Root found at: 3.71083
Number of iterations: 19
Process returned 0 (0x0)    execution time : 1.453 s
Press any key to continue.
```

By observation of the graph, n th positive root lies between $(n-1)$ and n . So choosing $[n-1, n]$ as the starting interval for bisection method gives the n th positive root as the answer. Similarly for the 25th root,

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen... - □ ×
Enter value of n: 25
Enter number of iterations: 200
Root found at: 24.5123
Number of iterations: 16
Process returned 0 (0x0)    execution time : 1.869 s
Press any key to continue.
```



3. SOURCE CODE

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  double getF(double x)
6  {
7      return (1000000*exp(x)) + ((435000/x)*(exp(x)-1)) - 1564000;
8  }
9
10 int main()
11 {
12     double prevVal;
13     double currentVal;
14     double firstInitial;
15     double secondInitial;
16     int n;
17     double epsilon = 0.000001;
18     bool rootFound = false;
19
20     cout<<getF(0.100998);
21     cout<<"Enter initial value: ";
22     cin>>firstInitial;
23     cout<<"Enter second initial value: ";
24     cin>>secondInitial;
25     prevVal = firstInitial;
26     currentVal = secondInitial;
27
28     cout<<"Enter number of iterations: ";
29     cin>>n;
30     cout<<"\n\nEVALUATING USING SECANTS METHOD:\n";
31

```

```

31
32     for(int i=0; i<n; i++)
33     {
34         double temp = currentVal;
35         currentVal = currentVal - ((getF(currentVal)*(currentVal-prevVal))/(getF(currentVal)-getF(prevVal)));
36         prevVal = temp;
37         if(abs(currentVal-prevVal)/abs(prevVal) < epsilon)
38         {
39             cout<<"Root found at: "<<currentVal;
40             cout<<"\nNumber of iterations: "<<i+1;
41             rootFound = true;
42             break;
43         }
44         cout<<prevVal<<endl;
45     }
46
47     if(!rootFound)
48         cout<<"Failed to converge!";
49
50
51     return 0;
52 }

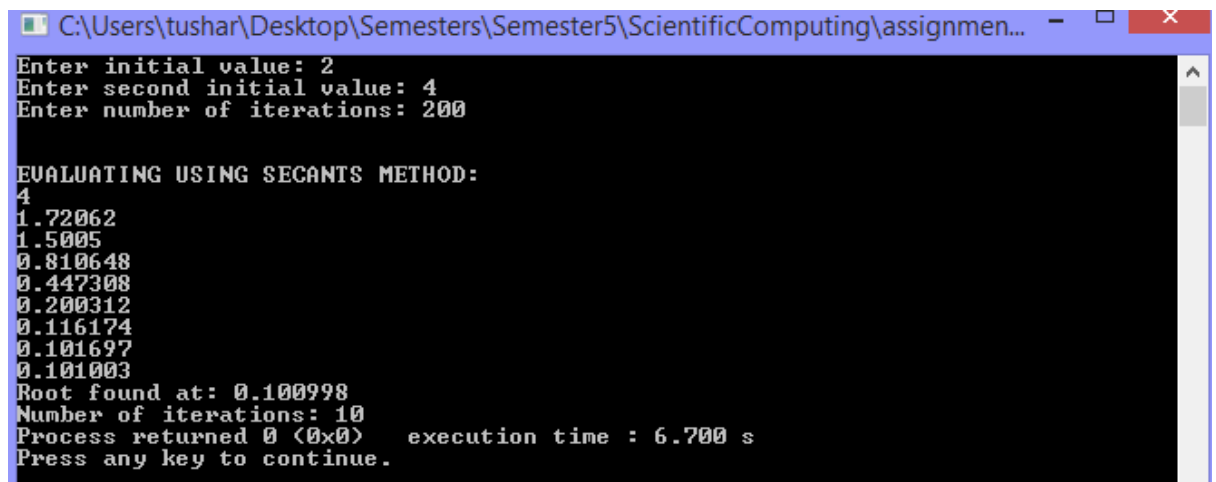
```

SECANT METHOD

Initial Values: 2, 4

Root Found: 0.100998

Number Of Iterations: 10



```

C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...
Enter initial value: 2
Enter second initial value: 4
Enter number of iterations: 200

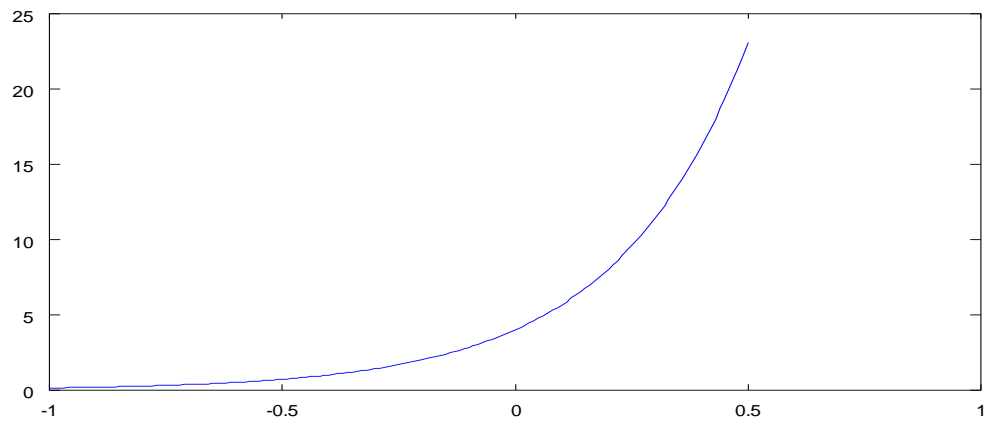
EVALUATING USING SECANTS METHOD:
4
1.72062
1.5005
0.810648
0.447308
0.200312
0.116174
0.101697
0.101003
Root found at: 0.100998
Number of iterations: 10
Process returned 0 (0x0)    execution time : 6.700 s
Press any key to continue.

```


4. SOURCE CODE

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5
6  double getF(double x)
7  {
8      return pow(3, (3*x)+1) - pow(7.5, 2*x);
9  }
10
11 double getFDASH(double x)
12 {
13     return (log(3)*pow(3, (3*x)+2)) - (2*log(7.5)*pow(7.5, 2*x));
14 }
15
16 int main()
17 {
18     double prevVal;
19     double currentVal;
20     double firstInitial;
21     double secondInitial;
22     int n;
23     double epsilon = 0.0000000000000001;
24     bool rootFound = false;
25
26     cout<<"Enter initial value: ";
27     cin>>firstInitial;
28     prevVal = firstInitial;
29
30     cout<<"Enter number of iterations: ";
31     cin>>n;
32
33     cout<<"EVALUATING USING NEWTONS METHOD:\n";
34     for(int i=0; i<n; i++)
35     {
36         // double temp = currentVal;
37         currentVal = prevVal - (getF(prevVal)/getFDASH(prevVal));
38         if(abs(currentVal - prevVal) < epsilon)
39         {
40             cout<<"Root found at: "<<currentVal;
41             cout<<"\nNumber of iterations: "<<i+1;
42             rootFound = true;
43             break;
44         }
45         prevVal = currentVal;
46         cout<<prevVal<<endl;
47     }
48
49     if(!rootFound)
50         cout<<"Failed to converge!";
51
52     return 0;
53 }
54
```

Graph of given function:



NEWTONS METHOD

Initial Value: 0

Root Found: -226.346

Number Of Iterations: 747

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...
Enter initial value: 0
Enter number of iterations: 2000
EVALUATING USING NEWTONS METHOD:
Root found at: -226.346
Number of iterations: 747
Process returned 0 (0x0)    execution time : 6.272 s
Press any key to continue.
```

5. SOURCE CODE

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5
6  double getF(double x)
7  {
8      return (x+1)*(x+1)*(x+1)*(x-1);
9  }
10
11 double getFDASH(double x)
12 {
13     return 2*(x+1)*(x+1)*(x-1);
14 }
15
16 int main()
17 {
18     double prevVal;
19     double currentVal;
20     double firstInitial;
21     double secondInitial;
22     int n;
23     double epsilon = 0.0001;
24     bool rootFound = false;
25
26
27     cout<<"Enter initial value: ";
28     cin>>firstInitial;
29     prevVal = firstInitial;
30
31     cout<<"Enter number of iterations: ";
32     cin>>n;
33
34     cout<<"EVALUATING USING NEWTONS METHOD:\n";
35     for(int i=0; i<n; i++)
36     {
37         // double temp = currentVal;
38         currentVal = prevVal - (getF(prevVal)/getFDASH(prevVal));
39         if(abs(currentVal + 1) < epsilon)
40         {
41             cout<<"Root found at: "<<currentVal;
42             cout<<"\nNumber of iterations: "<<i+1;
43             rootFound = true;
44             break;
45         }
46         prevVal = currentVal;
47         cout<<prevVal<<endl;
48     }
49
50     if(!rootFound)
51         cout<<"Failed to converge!";
52
53     return 0;
54 }
55

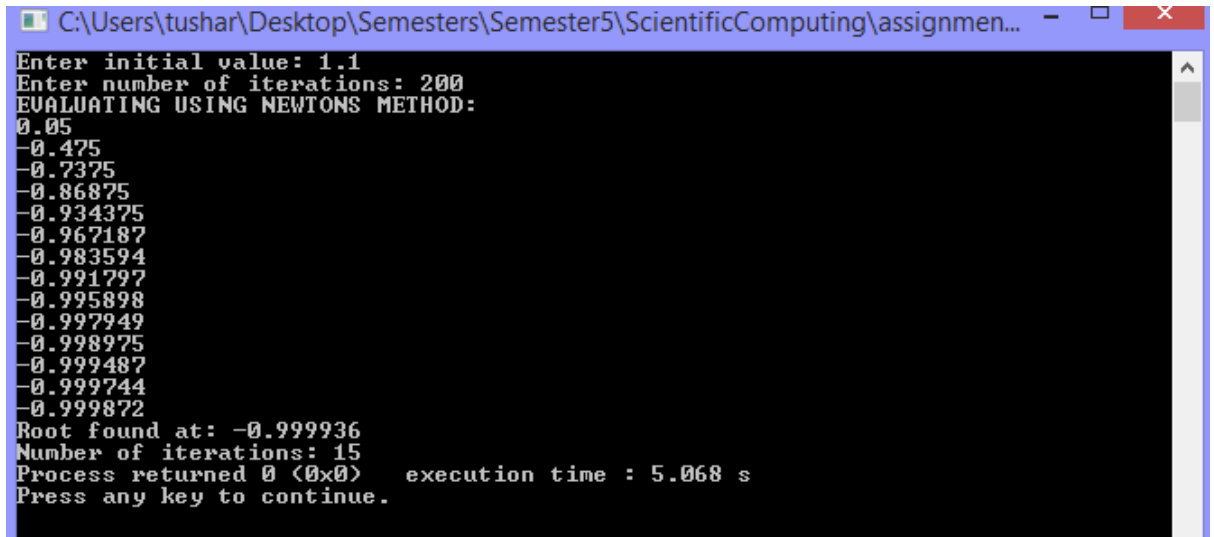
```

NEWTON METHOD

Initial Value: 1.1

Root Found: -0.999936

Number Of Iterations: 15

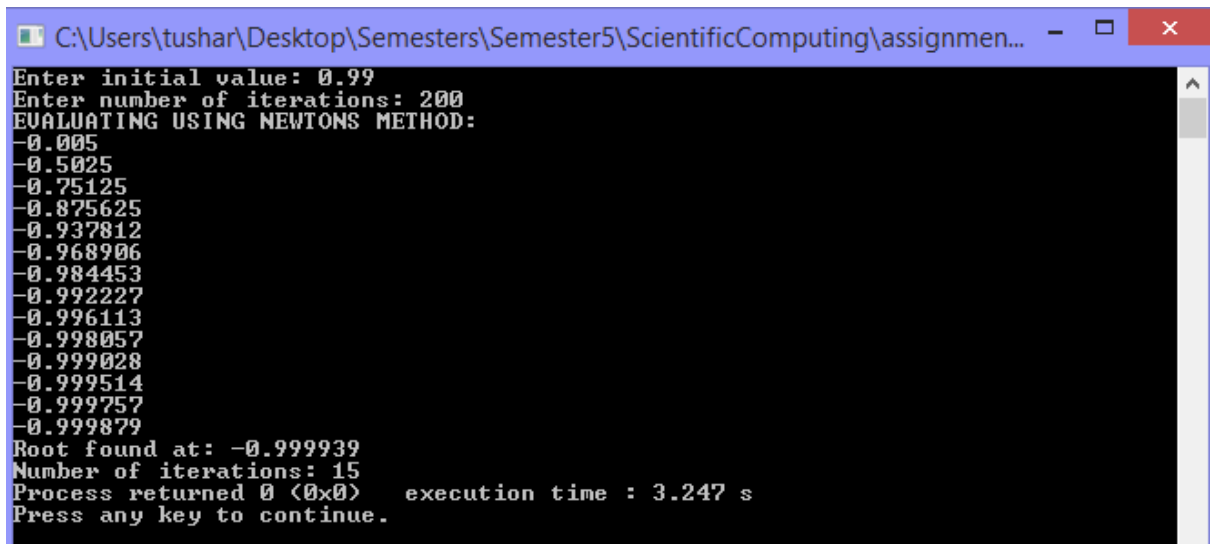


```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...  
Enter initial value: 1.1  
Enter number of iterations: 200  
EVALUATING USING NEWTONS METHOD:  
0.05  
-0.475  
-0.7375  
-0.86875  
-0.934375  
-0.967187  
-0.983594  
-0.991797  
-0.995898  
-0.997949  
-0.998975  
-0.999487  
-0.999744  
-0.999872  
Root found at: -0.999936  
Number of iterations: 15  
Process returned 0 (0x0)   execution time : 5.068 s  
Press any key to continue.
```

Initial Value: 0.99

Root Found: -0.999939

Number Of Iterations: 15



```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...  
Enter initial value: 0.99  
Enter number of iterations: 200  
EVALUATING USING NEWTONS METHOD:  
-0.005  
-0.5025  
-0.75125  
-0.875625  
-0.937812  
-0.968906  
-0.984453  
-0.992227  
-0.996113  
-0.998057  
-0.999028  
-0.999514  
-0.999757  
-0.999879  
Root found at: -0.999939  
Number of iterations: 15  
Process returned 0 (0x0)   execution time : 3.247 s  
Press any key to continue.
```

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  double getF(double x)
6  {
7      return (x+1)*(x+1)*(x+1)*(x-1);
8  }
9
10 double getFDASH(double x)
11 {
12     return 2*(x+1)*(x+1)*(x-1);
13 }
14
15 int main()
16 {
17     double prevVal;
18     double currentVal;
19     double firstInitial;
20     double secondInitial;
21     int n;
22     double epsilon = 0.0001;
23     bool rootFound = false;
24
25     cout<<"Enter initial value: ";
26     cin>>firstInitial;
27     cout<<"Enter second initial value: ";
28     cin>>secondInitial;
29     prevVal = firstInitial;
30     currentVal = secondInitial;
31

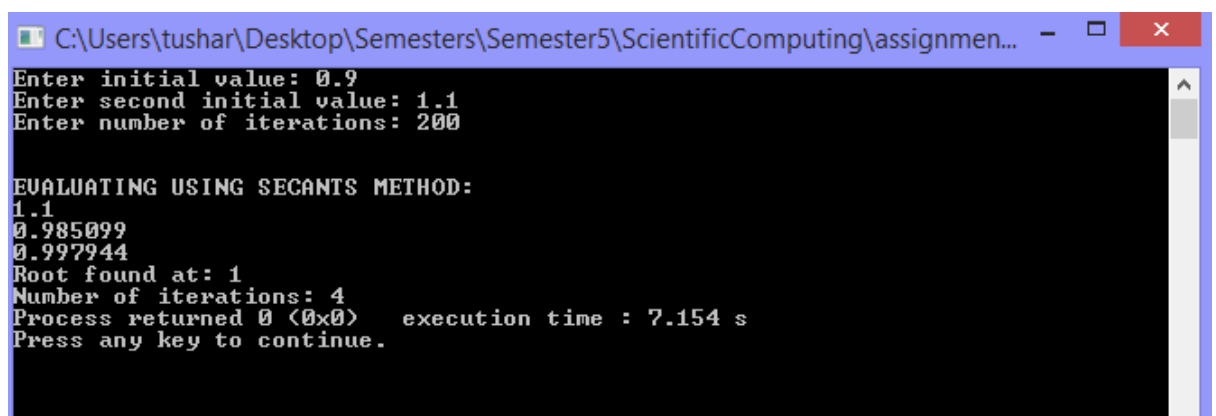
```

SECANT METHOD

Initial Values: 0.9, 1.1

Root Found: 1

Number Of Iterations: 4



```

C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...
Enter initial value: 0.9
Enter second initial value: 1.1
Enter number of iterations: 200

EVALUATING USING SECANTS METHOD:
1.1
0.985099
0.997944
Root found at: 1
Number of iterations: 4
Process returned 0 (0x0)   execution time : 7.154 s
Press any key to continue.

```