University of Dhaka

Department of Computer Science and Engineering

CSE-3212: Numerical Methods Lab
3rd Year 2nd Semester

Assignment: 02

Problems on Bisection, False Position,
Newton-Raphson and Secant methods

Submitted by:

Amit Sarker, Roll: 99

Submitted to:

Mr. Mubin Ul Haque, Lecturer, Department of CSE, University of Dhaka

Date of submission: 14th September, 2018.

Problem 1:

Statement:

The velocity v of a falling parachutist is given by

$$v = \frac{gm}{c}(1 - e^{-(\frac{c}{m})t})$$

where g = 9.8 m/s 2 . For a parachutist with a drag coefficient c = 15 kg/s, compute the mass m so that the velocity is v = 35 m/s at t = 9 s.

By using

(a) bisection

and (b) false position.

Solution:

Source Code

Ideone Link: https://ideone.com/5Gps73

```
#include<bits/stdc++.h>
using namespace std;
typedef long long Long;
double a, b, acc;
const double eps = 1e-9;

double f(double x)
{
    return ((9.8*x)/15.0) * (1-exp((-15.0/x) * 9.0)) - 35.0;
}

double relativeError(double newxm, double oldxm)
{
    return fabs(((newxm - oldxm) / newxm) * 100);
}

void printfunc(FILE *f1, FILE *f2, int iteration, double upper,
double lower, double newxm, double oldxm, double fxm)
{
    char l1[11]="Iter";
    char l2[11]="Upper";
    char l3[11]="Lower";
    char l4[11]="Xm";
    char l5[11]="f(Xm)";
    char l6[11]="Error";
```

```
    if(iteration == 1)
    {
        printf("--------------------------------Bisection
Method-------------------------------------\n");
        printf("|%12s |%12s |%12s |%12s | %12s|
%12s|\n",l1,l2,l3,l4,l5,l6);

printf("|-------------|-------------|-------------|-------------|----
---------|-------------|\n");
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf | %-12s|\n",
iteration, upper, lower, newxm, fxm, "N/A");

    }
    else
    {
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf | %-12lf|\n",
iteration, upper, lower, newxm, fxm, relativeError(newxm, oldxm));
        fprintf(f1, "%d,%lf\n", iteration, relativeError(newxm,
oldxm) * 100);
        fprintf(f2, "%lf,%lf\n", newxm, relativeError(newxm, oldxm) *
100);
    }
}

void printfunc_false(FILE *f_false1, FILE *f_false2, int iteration,
double upper, double lower, double newxm, double oldxm, double fxm)
{
    char l1[11]="Iter";
    char l2[11]="Upper";
    char l3[11]="Lower";
    char l4[11]="Xm";
    char l5[11]="f(Xm)";
    char l6[11]="Error(\%)";
    if(iteration == 1)
    {
        printf("--------------------------------False Position
Method-------------------------------\n");
        printf("|%12s |%12s |%12s |%12s | %12s|
%12s|\n",l1,l2,l3,l4,l5,l6);

printf("|-------------|-------------|-------------|-------------|----
---------|-------------|\n");
```

```
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf | %-12s|\n",
iteration, upper, lower, newxm, fxm, "N/A");

    }
    else
    {
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf | %-12lf|\n",
iteration, upper, lower, newxm, fxm, relativeError(newxm, oldxm));
        fprintf(f_false1, "%lf,%lf\n", newxm, relativeError(newxm,
oldxm) * 100);
        fprintf(f_false2, "%d,%lf\n", iteration, relativeError(newxm,
oldxm) * 100);
    }
}

void regulaFalsi(double a, double b)
{
    if (f(a) * f(b) >= 0)
    {
        printf("Wrong assumption of a and b (False Position)\n");
        return;
    }

    double c = a;
    int i = 1;
    double oldxm = 0.0, newxm;
    FILE *f_false1 = fopen("g3_error3_false_XvsE.csv", "w");
    FILE *f_false2 = fopen("g4_error4_false_IvsE.csv", "w");
    puts("");

    while (1)
    {
        oldxm = c;
        c = (a*f(b) - b*f(a))/ (f(b) - f(a));
        newxm = c;
        printfunc_false(f_false1, f_false2, i, a, b, newxm, oldxm,
f(newxm));

        if(relativeError(newxm, oldxm) <= acc) break;
        if (f(c)==0) break;
        else if (f(c)*f(a) < 0) b = c;
        else a = c;
        i++;
```

```
    }
    printf("\nThe value of root is (False Position) : %lf\n",c);
}

void bisection(double a, double b)
{
    puts("");
    cout<<"Function values for the initial guesses: "<<endl;
    cout<<"For "<<a<<": "<<f(a)<<"       "<<"For "<<b<<":
"<<f(b)<<endl;
    puts("");
    if (f(a) * f(b) >= 0)
    {
        printf("\nWrong assumption of a and b (Bisection)\n");
        return;
    }
    puts("");

    FILE *f1 = fopen("g2_error1_bisection_IvsE.csv", "w");
    FILE *f2 = fopen("g3_error2_bisection_XvsE.csv", "w");
    double c = a;
    int i = 1;
    double oldxm = 0.0, newxm;
    while(1)
    {
        oldxm = c;
        c = (a+b)/2;
        newxm = c;
        printfunc(f1, f2, i, b, a, newxm, oldxm, f(newxm));

        if(relativeError(newxm, oldxm) <= acc) break;
        if (f(c) == 0.0) break;
        else if (f(c) * f(a) < 0) b = c;
        else a = c;
        i++;
    }
    printf("\nThe value of root is (Bisection) : %lf\n",c);
}

int main()
{
    cout<<"Enter the value of initial guesses & Accuracy: "<<endl;
    cin>>a>>b>>acc;
```

```
    FILE *fp1 = fopen("g1_bisection.csv", "w");
    printf("\n   X                          f(X)\n\n");
    for(double i = a; i <= b+eps; i += 0.1)
    {
        fprintf(fp1, "%lf,%lf\n", i, f(i));
        printf("%lf                   %lf\n", i, f(i));
    }
    bisection(a, b);
    regulaFalsi(a, b);
    return 0;
}
```

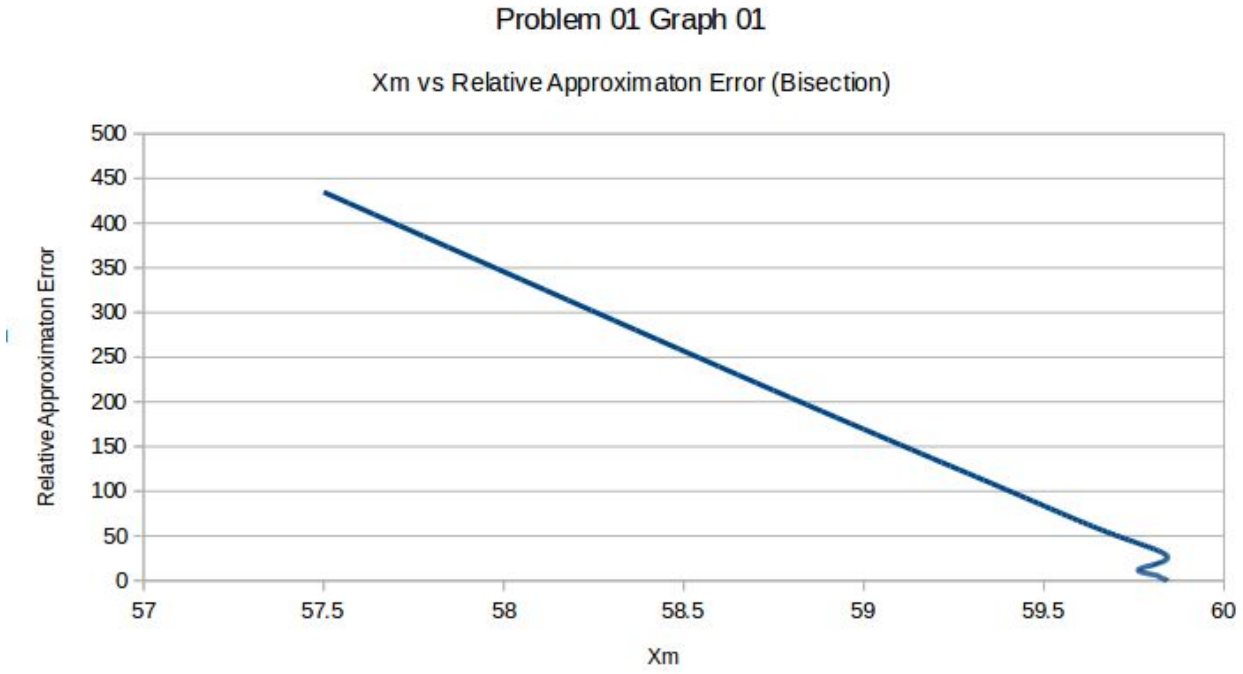
Sample Input/Output (Console View):

```
Enter the value of initial guesses & Accuracy:
58 60 0.00001

   X                          f(X)

58.000000                   -0.802437
58.100000                   -0.758337
58.200000                   -0.714296
58.300000                   -0.670315
58.400000                   -0.626393
58.500000                   -0.582530
58.600000                   -0.538726
58.700000                   -0.494982
58.800000                   -0.451296
58.900000                   -0.407670
59.000000                   -0.364102
59.100000                   -0.320594
59.200000                   -0.277144
59.300000                   -0.233752
59.400000                   -0.190419
59.500000                   -0.147145
59.600000                   -0.103929
59.700000                   -0.060772
59.800000                   -0.017673
59.900000                   0.025368
60.000000                   0.068350

Function values for the initial guesses:
For 58: -0.802437       For 60: 0.0683504
```

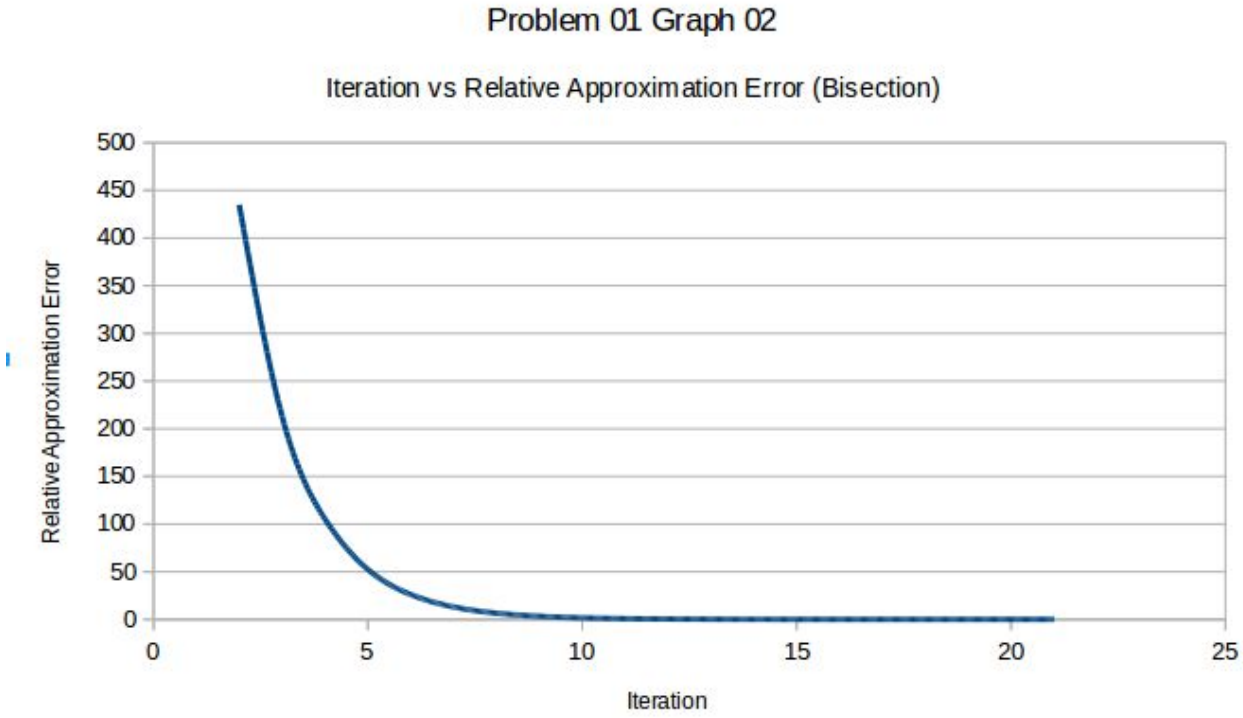
```
--------------------------------Bisection Method-------------------------------------
|        Iter |       Upper |       Lower |          Xm |        f(Xm)|        Error|
|-------------|-------------|-------------|-------------|-------------|-------------|
|1            |60.000000    |58.000000    |59.000000    |-0.364102    | N/A         |
|2            |60.000000    |59.000000    |59.500000    |-0.147145    | 0.840336    |
|3            |60.000000    |59.500000    |59.750000    |-0.039215    | 0.418410    |
|4            |60.000000    |59.750000    |59.875000    |0.014613     | 0.208768    |
|5            |59.875000    |59.750000    |59.812500    |-0.012290    | 0.104493    |
|6            |59.875000    |59.812500    |59.843750    |0.001164     | 0.052219    |
|7            |59.843750    |59.812500    |59.828125    |-0.005562    | 0.026116    |
|8            |59.843750    |59.828125    |59.835938    |-0.002199    | 0.013057    |
|9            |59.843750    |59.835938    |59.839844    |-0.000517    | 0.006528    |
|10           |59.843750    |59.839844    |59.841797    |0.000324     | 0.003264    |
|11           |59.841797    |59.839844    |59.840820    |-0.000097    | 0.001632    |
|12           |59.841797    |59.840820    |59.841309    |0.000114     | 0.000816    |
|13           |59.841309    |59.840820    |59.841064    |0.000008     | 0.000408    |
|14           |59.841064    |59.840820    |59.840942    |-0.000044    | 0.000204    |
|15           |59.841064    |59.840942    |59.841003    |-0.000018    | 0.000102    |
|16           |59.841064    |59.841003    |59.841034    |-0.000005    | 0.000051    |
|17           |59.841064    |59.841034    |59.841049    |0.000002     | 0.000025    |
|18           |59.841049    |59.841034    |59.841042    |-0.000001    | 0.000013    |
|19           |59.841049    |59.841042    |59.841045    |0.000000     | 0.000006    |

The value of root is (Bisection) : 59.841045

--------------------------------False Position Method-------------------------------
|        Iter |       Upper |       Lower |          Xm |        f(Xm)|     Error(%)|
|-------------|-------------|-------------|-------------|-------------|-------------|
|1            |58.000000    |60.000000    |59.843015    |0.000848     | N/A         |
|2            |58.000000    |59.843015    |59.841069    |0.000011     | 0.003251    |
|3            |58.000000    |59.841069    |59.841045    |0.000000     | 0.000040    |
|4            |58.000000    |59.841045    |59.841045    |0.000000     | 0.000000    |

The value of root is (False Position) : 59.841045

Process returned 0 (0x0)   execution time : 7.479 s
Press ENTER to continue.
```

Graphs: (Next Page):

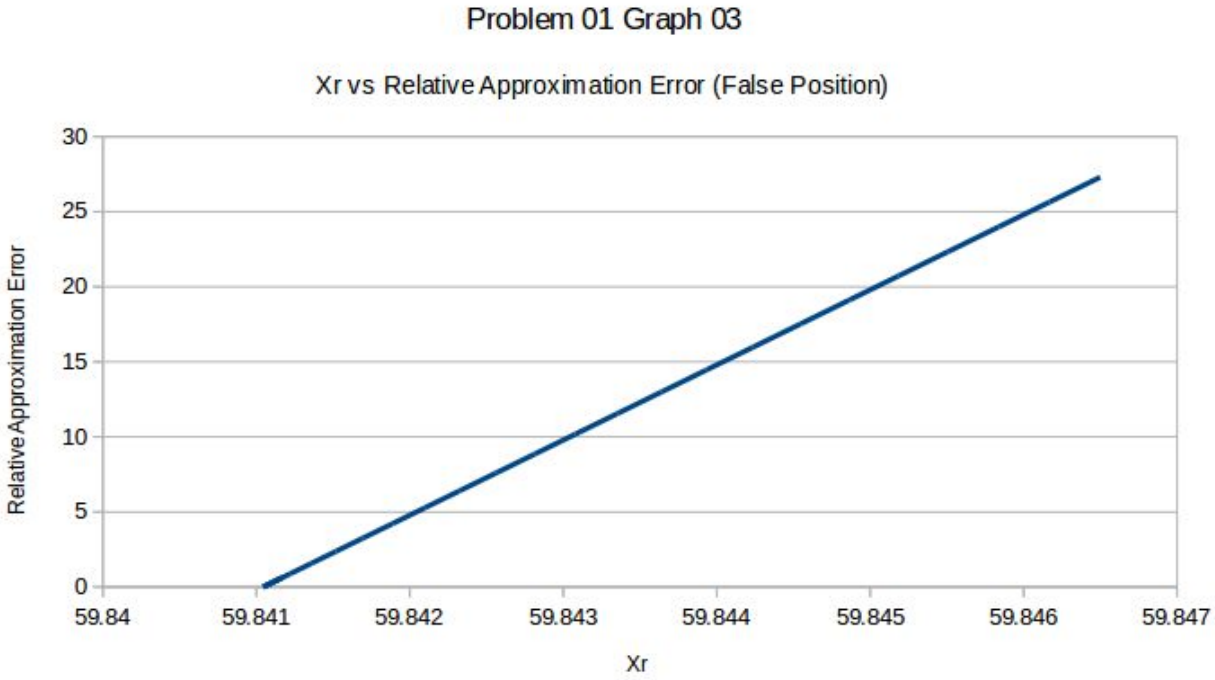
Graph 01: The graph of Xm and relative approximation error (bisection).



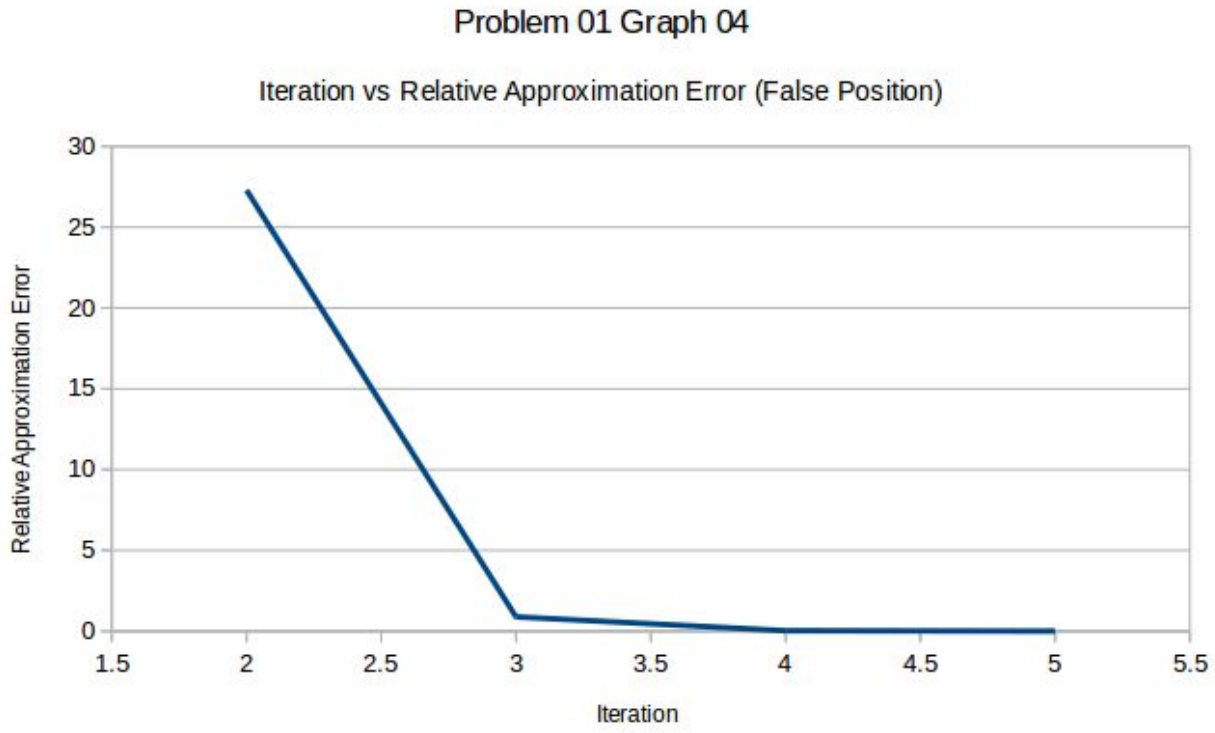
Graph 02: The graph of no of iteration and relative approximation error (bisection).



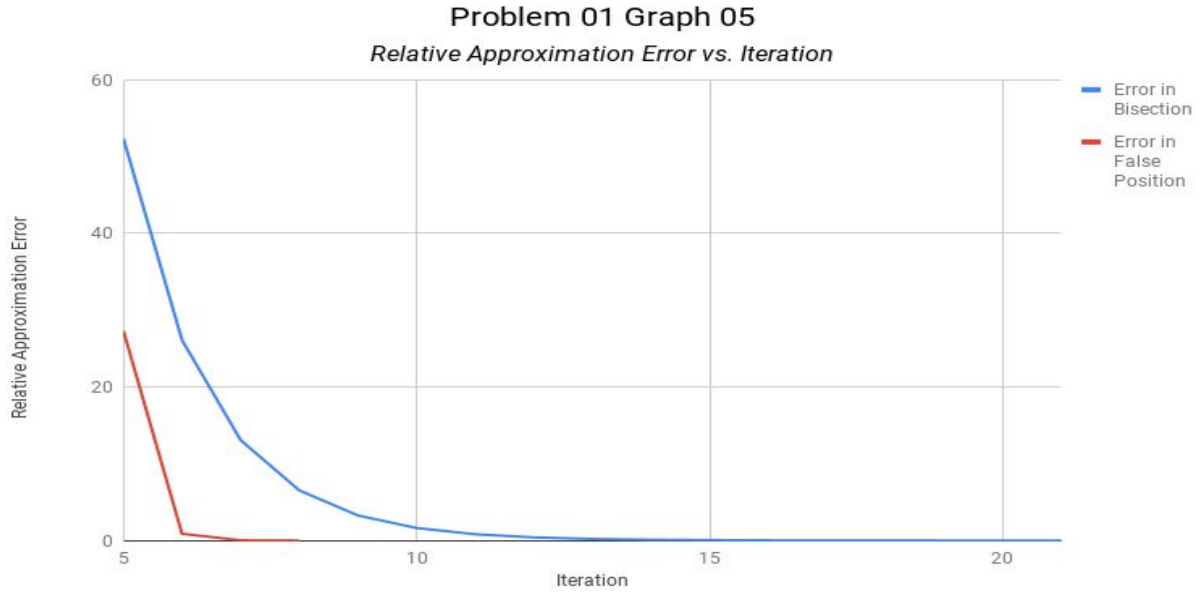
Graph 03: The graph of Xr and relative approximation error (false position).



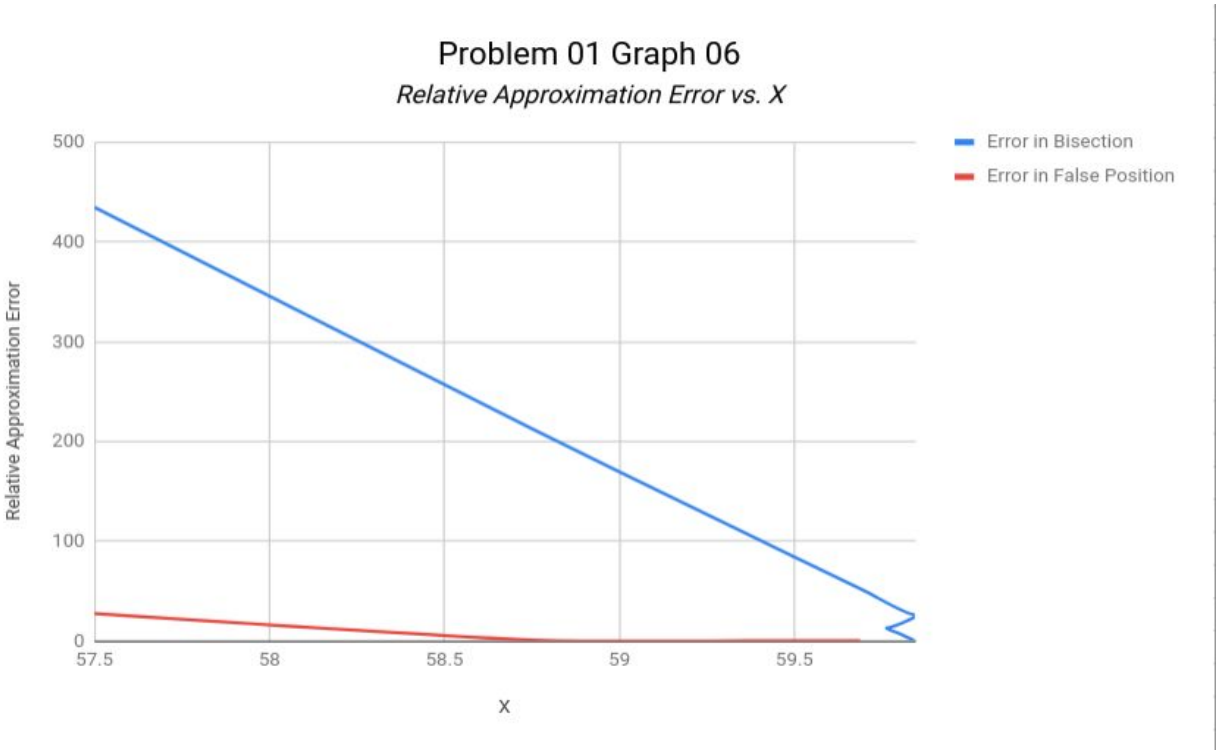
Graph 04: The graph of no of iteration and relative approximation error (false position).



Graph 05: Comparison of Relative approximate error with respect to iteration between the bisection method and false position method.



Graph 06: Comparison of Relative approximate error with respect to X between the bisection method and false position method.



Problem 2:

Statement:

Write a single program (source file name must be problem2. extension) to solve the following
(a) Use the Newton-Raphson method to determine a root of $f(x) = -x^2 + 1.8x + 2.5$ using $x_0 = 5$.
Perform the computation until εa is less than user specified tolerance.
Also perform an error check of your final answer as the following table.
(b) Use the Newton-Raphson method to find the root of
$f(x) = e^{-0.5x}(4 - x) - 2$
Employ initial guesses of (i) 2, (ii) 6, and (iii) 8. Explain your
Results.

Solution:

Source Code

Ideone Link: https://ideone.com/AhvYTT

```
#include<bits/stdc++.h>
using namespace std;
double acc;

double f(double x, int f_choice)
{
    if(!f_choice) return -x*x+1.8*x+2.5;
    else return exp(-0.5*x)*(4-x)-2;
}

double f_prime(double x, int f_choice)
{
    if(!f_choice) return -2*x+1.8;
    else return -0.5*exp(-0.5*x)*(4-x)-exp(-0.5*x);
}

double rootRaphson(double x_val, double f_xval, int f_choice)
{
    double x = f_xval/f_prime(x_val, f_choice);
    return x_val-x;
}

double relativeError(double newxm, double oldxm)
```

```
{
    return fabs(((newxm - oldxm) / newxm));
}


void printfunc(int iteration, double x_val, double f_xval, double
f_prime_xval, double newroot, double oldroot)
{
    char l1[11]="Iter";
    char l2[11]="X_val";
    char l3[11]="f(X_val)";
    char l4[11]="f'(X_val)";
    char l5[11]="New_root";
    char l6[11]="Error";
    if(iteration == 1)
    {
        printf("--------------------------------Newton Raphson
Method--------------------------------\n");
        printf("|%12s |%12s |%12s |%12s | %12s|
%12s|\n",l1,l2,l3,l4,l5,l6);

printf("|-------------|-------------|-------------|-------------|----
---------|-------------|\n");
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf | %-12s|\n",
iteration, x_val, f_xval, f_prime_xval, newroot, "N/A");

    }
    else
    {
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf | %-12lf|\n",
iteration, x_val, f_xval, f_prime_xval, newroot,
relativeError(newroot, oldroot));
    }
}

void newtonRaphson(double x_val, int f_choice)
{
    int i = 1;
    double error, oldroot, newroot = x_val;
    while(1)
    {
        double f_xval = f(x_val, f_choice);
        double f_prime_xval = f_prime(x_val, f_choice);
```

```
        oldroot = x_val;
        newroot = rootRaphson(x_val, f_xval, f_choice);

        printfunc(i, x_val, f_xval, f_prime_xval, newroot, oldroot);
        error = relativeError(newroot, oldroot);

        if(error <= acc) break;
        if(f(newroot, f_choice) == 0.0) break;

        x_val = newroot;
        i++;
    }
    printf("\nThe value of root is (Newton-Raphson) : %.8lf\n",
newroot);
}
int main()
{
    double x_val;
    int f_choice;

    cout<<"Choose which part of the problem you want to solve: 0 for
(a), 1 for (b)"<<endl;
    cin>>f_choice;
    if(!f_choice)
    {
        cout<<"Enter the value of initial guess & Accuracy: "<<endl;
        cin>>x_val>>acc;
        newtonRaphson(x_val, f_choice);
    }
    else
    {
        for(int i = 1; i <= 3; i++)
        {
            cout<<"Enter the value of initial guess "<<i<<" &
Accuracy: "<<endl;
            cin>>x_val>>acc;
            newtonRaphson(x_val, f_choice);
        }
    }
    puts("\n\nRunning Again...\n\n");
    main();
    return 0;
}
```

Sample Input/Output (Console View):

```
Choose which part of the problem you want to solve: 0 for (a), 1 for (b)
0
Enter the value of initial guess & Accuracy:
5  0.00001
---------------------------------Newton Raphson Method---------------------------------
|        Iter |       X_val |    f(X_val) |   f'(X_val) |     New_root|        Error|
|-------------|-------------|-------------|-------------|-------------|-------------|
|1            |5.000000     |-13.500000   |-8.200000    |3.353659     | N/A         |
|2            |3.353659     |-2.710440    |-4.907317    |2.801332     | 0.197166    |
|3            |2.801332     |-0.305064    |-3.802665    |2.721108     | 0.029482    |
|4            |2.721108     |-0.006436    |-3.642217    |2.719341     | 0.000650    |
|5            |2.719341     |-0.000003    |-3.638683    |2.719341     | 0.000000    |

The value of root is (Newton-Raphson) : 2.71934054


Running Again...


Choose which part of the problem you want to solve: 0 for (a), 1 for (b)
1
Enter the value of initial guess 1 & Accuracy:
2  0.00001
---------------------------------Newton Raphson Method---------------------------------
|        Iter |       X_val |    f(X_val) |   f'(X_val) |     New_root|        Error|
|-------------|-------------|-------------|-------------|-------------|-------------|
|1            |2.000000     |-1.264241    |-0.735759    |0.281718     | N/A         |
|2            |0.281718     |1.229743     |-2.483483    |0.776887     | 0.637376    |
|3            |0.776887     |0.185630     |-1.770927    |0.881708     | 0.118884    |
|4            |0.881708     |0.006579     |-1.646776    |0.885703     | 0.004511    |
|5            |0.885703     |0.000009     |-1.642207    |0.885709     | 0.000006    |

The value of root is (Newton-Raphson) : 0.88570880
Enter the value of initial guess 2 & Accuracy:
4  0.00001
---------------------------------Newton Raphson Method---------------------------------
|        Iter |       X_val |    f(X_val) |   f'(X_val) |     New_root|        Error|
|-------------|-------------|-------------|-------------|-------------|-------------|
|1            |4.000000     |-2.000000    |-0.135335    |-10.778112   | N/A         |
|2            |-10.778112   |3234.355984  |-1837.174569 |-9.017607    | 0.195230    |
|3            |-9.017607    |1180.168964  |-681.897563  |-7.286894    | 0.237510    |
|4            |-7.286894    |429.422999   |-253.934860  |-5.595818    | 0.302203    |
|5            |-5.595818    |155.470248   |-95.145423   |-3.961791    | 0.412447    |
|6            |-3.961791    |55.716856    |-36.107659   |-2.418715    | 0.637973    |
|7            |-2.418715    |19.511234    |-14.106947   |-1.035621    | 1.335522    |
|8            |-1.035621    |6.451528     |-5.904113    |0.057097     | 19.137991   |
|9            |0.057097     |1.831931     |-2.887821    |0.691461     | 0.917426    |
|10           |0.691461     |0.341463     |-1.878435    |0.873242     | 0.208168    |
|11           |0.873242     |0.020562     |-1.656497    |0.885655     | 0.014016    |
|12           |0.885655     |0.000088     |-1.642262    |0.885709     | 0.000061    |
|13           |0.885709     |0.000000     |-1.642201    |0.885709     | 0.000000    |

The value of root is (Newton-Raphson) : 0.88570880
```

Problem 2(b) Discussion:

In problem 2(b), there are three initial guesses, 2, 6 and 8. For initial guess 2, Newton Raphson can calculate the root for the given function and the root value is 0.88570880. I have attached the console view in the previous page. Not only for initial guess 2, actually Newton Raphson can calculate the root value when initial guess is less than 6. I have calculated for initial guess 2 and 4. But when initial guess becomes greater than 6, the function value of the derivative of the given function becomes 0. For this, when i tried to calculate the root, the equation for Newton Raphson $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ returns infinity as $f'(x_i)$ becomes 0. For this, Newton Raphson can not calculate the root value for the given function when initial guess is 6 and 8. The graph of the derivative function of the given function is shown below. We can see from the graph that, when x >= 6, the function value is equal to 0.

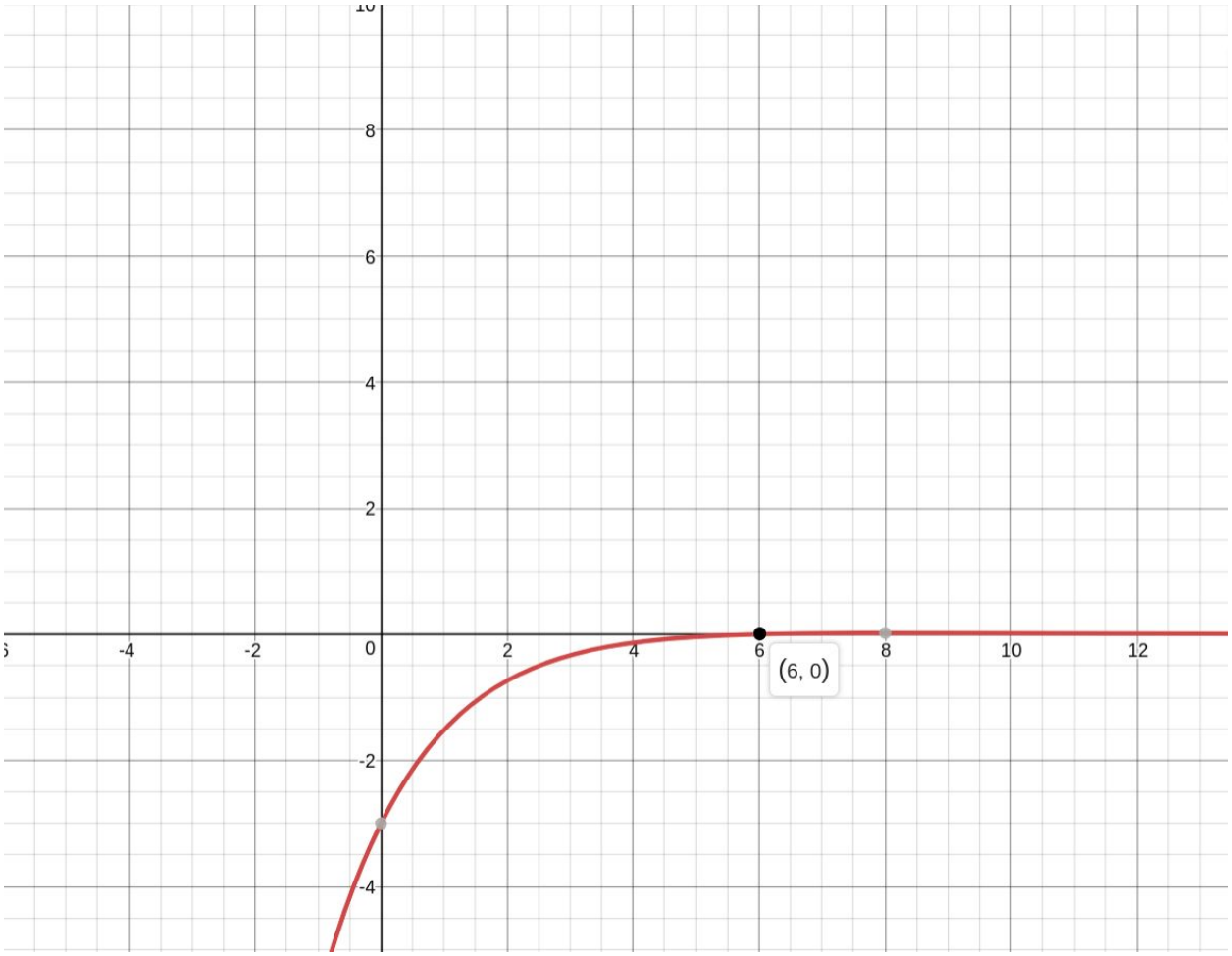


Fig: Graph of the function $-0.5e^{-0.5x}(4-x) - e^{-0.5x}$ which is the derivative of the given function $e^{-0.5x}(4-x) - 2$.

Problem 3:

Statement:

Write a single program (source file name must be problem3. extension) to solve the following
(a) Consider following easily differentiable function,
$f(x) = 8sin(x)e^{-x} - 1$
Use the secant method, when initial guesses of $x_{i-1} = 0.5$ and $x_i = 0.4$ with user specified Tolerance.

Solution:

Source Code

Ideone Link: https://ideone.com/aiArGa

```
#include<bits/stdc++.h>
using namespace std;
double acc;

double f(double x)
{
    return 8*sin(x)*exp(-x)-1;
}

double rootSacant(double lower, double upper, double f_lower, double
f_upper)
{
    double x = (f_upper*(upper-lower))/(f_upper-f_lower);
    return upper-x;
}

double relativeError(double newxm, double oldxm)
{
    return fabs(((newxm - oldxm) / newxm));
}

void printfunc(int iteration, double lower, double upper, double
f_lower, double f_upper, double newroot, double oldroot)
{
    char l1[11]="Iter";
    char l2[11]="Lower";
    char l3[11]="Upper";
```

```
    char l4[11]="f(Lower)";
    char l5[11]="f(Upper)";
    char l6[11]="New_root";
    char l7[11]="Error";
    if(iteration == 1)
    {
        printf("-----------------------------------------Secant
Method-----------------------------------------\n");
        printf("|%12s |%12s |%12s |%12s | %12s| %12s|
%12s|\n",l1,l2,l3,l4,l5,l6,l7);

printf("|-------------|-------------|-------------|-------------|----
---------|-------------|-------------|\n");
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf |%-12lf |
%-12s|\n", iteration, lower, upper, f_lower, f_upper, newroot,
"N/A");

    }
    else
    {
        printf("|%-12d |%-12lf |%-12lf |%-12lf |%-12lf |%-12lf |
%-12lf|\n", iteration, lower, upper, f_lower, f_upper, newroot,
relativeError(newroot, oldroot));
    }
}

void secant(double lower, double upper)
{
    int i = 1;
    double error, oldroot, newroot = lower;
    while(1)
    {
        double f_lower = f(lower);
        double f_upper = f(upper);
        oldroot = newroot;
        newroot = rootSacant(lower, upper, f_lower, f_upper);

        printfunc(i, lower, upper, f_lower, f_upper, newroot,
oldroot);

        if(i!=1)
        {
            error = relativeError(newroot, oldroot);
```

```
            if(error <= acc) break;
        }

        if (f(newroot) == 0.0) break;

        lower = upper;
        upper = newroot;

        i++;
    }
    printf("\nThe value of root is (Secant) : %.8lf\n", newroot);
}

int main()
{
    double lower, upper, root;
    cout<<"Enter the value of initial guess & Accuracy: "<<endl;
    cin>>lower>>upper>>acc;
    if(lower>upper) swap(lower, upper);
    secant(lower, upper);
    puts("\n\nRunning Again...\n\n");
    main();
    return 0;
}
```

Sample Input/Output (Console View):

```
Enter the value of initial guess & Accuracy:
0.4  0.5  0.00001
-----------------------------------------Secant Method-----------------------------------------
|        Iter |       Lower |       Upper |    f(Lower) |     f(Upper)|     New_root|        Error|
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|1            |0.400000     |0.500000     |1.088279     |1.326290     |-0.057239    | N/A         |
|2            |0.500000     |-0.057239    |1.326290     |-1.484624    |0.237075     | 1.241440    |
|3            |-0.057239    |0.237075     |-1.484624    |0.482310     |0.164906     | 0.437633    |
|4            |0.237075     |0.164906     |0.482310     |0.113625     |0.142665     | 0.155901    |
|5            |0.164906     |0.142665     |0.113625     |-0.013780    |0.145070     | 0.016583    |
|6            |0.142665     |0.145070     |-0.013780    |0.000325     |0.145015     | 0.000382    |
|7            |0.145070     |0.145015     |0.000325     |0.000001     |0.145015     | 0.000001    |

The value of root is (Secant) : 0.14501481


Running Again...
```