CSE2202: Numerical Methods Lab
Online: 3 Section: B2
Time: 40 Minutes Total: 10

Note:
2 sets of problems are given for this online lab test.
Set A: All Possible Roots by Modified Bisection Method
Set B: Multiple Roots of Polynomial using Newtons Method.

- If you choose Set A you will get 20% penalty and for choosing Set B you will get no penalty.
- After completing your code you must upload you code and output in the given Google form link.
- Allocated time for Set A is 30 minutes and for Set C is 40 Minutes

Set A

Problem Statement: Determine the all possible real roots of the equation: $f(x) = x^3 - 7x^2 + 15x - 9 = 0$ using Modified Bisection Method. Employ initial guesses of x_{lower} = 0 and x_{upper} = 4 and iterate until the estimated error $€_a$ falls below a level of $€_s$ = 0.001

Algorithm:

1. Enter lower limit x_{lower} and upper limit x_{upper} of the interval covering all the roots.
2. Decide the size of the increment interval $\Delta x = 0.1$
3. set $x_1 = x_{lower}$ and $x_2 = x_{lower} + \Delta x$
4. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$
5. If $(f_1 * f_2)>0$,
   then the interval does not bracket any root and go to step 9
6. Compute $x_0 = (x_1 + x_2)/2$ and $f_0 = f(x_0)$
7. If $(f_1 * f_2)<0$
   then set $x_2 = x_0$
   Else set $x_1 = x_0$ and $f_1 = f_0$
8. If $|(x_2 - x_1)/x_2| < E$, then
   root = $(x_1+x_2) / 2$
   write the value of root
   go to step 9
   Else
   go to step 6
9. If $x_2 < x_{upper}$, then set $x_{lower} = x_2$ and go to step 3
10. Stop.

Tasks:

1. Write a program using Modified Bisection Method to locate the approximate roots of the function $f(x) = x^3 - 7x^2 + 15x - 9 = 0$ with initial guesses [0, 4].
2. Use Horner's rule to perform all iterations of the Modified Bisection Method until the estimated error $€_a$ falls below a level of $€_s = 0.0001$
3. Use appropriate functions from math header file.
4. Show the table with number of root, approximate value; number of iterations where the root is found and relative error found on that iteration. (You can count the step by adding 1 to a counting variable *i* in the loop of the program).

Sample Input/output:

```
Enter the highest degree of the equation: 3

Enter values of coefficients:
Coefficient x[3] = 1

Coefficient x[2] = -7

Coefficient x[1] = 15

Coefficient x[0] = -9
Enter the lower and upper limit:
0
4
Number of Root   Approximate Root        Number of Iteration       Relative Error
1                 0.999994                9                        0.000006
2                 2.999682                29                       0.000008
3                 3.000341                30                       0.000008
```

Set B

Problem Statement: Determine the **multiple** real roots of the equation: $f(x) = x^3 - 4x^2 + x + 6 = 0$ using Newton's Method. Employ initial guess of $x_1 = 5$ and iterate until the estimated error $€_a$ falls below a level of $€_s = 0.001$

Algorithm:

1. Obtain degree and co-efficient of polynomial (n and a_i).
2. Decide an initial estimate for the first root (x_0) and error criterion, E.
   Do while n >1
3. Find the root using Newton-Raphson algorithm
   $x_r = x_0 - f(x_0) / f'(x_0)$
4. Root (n) = x_r
5. Deflate the polynomial using synthetic division algorithm and make the factor polynomial as the new polynomial of order n-1.
6. Set $x_0 = x_r$ [Initial value of the new root]
   End of Do
7. Root (1) = $-\,a0 \,/\, a1$
8. Stop

Tasks:

1. Write a program using Newton's Method to locate the approximate roots of the function $(x) = x^3 - 4x^2 + x + 6 = 0$ with initial guess 5.
2. Use Horner's rule to perform all iterations of the Newton's Method until the estimated error $€_a$ falls below a level of $€_s = 0.001$
3. Use synthetic division to deflate the polynomial at lower degree.
   Algorithm for Synthetic Division:
   $b_{i-1} = a_i + x_r b_i \qquad ; for\ i = n, n-1, \ldots .0$
   $b_n = 0$

   Where a is the coefficient at degree n and b is the coefficient at degree $n - 1$
4. Use appropriate functions from math header file.
5. Print the degree of the polynomial, polynomial and roots found at each degree.

Sample Input/output:

```
 Enter values of coefficients:
 Coefficient x[3] = 1

 Coefficient x[2] = -4

 Coefficient x[1] = 1

 Coefficient x[0] = 6

Enter the intial value:5
The 3 order polynomial is:     1.00x^3+-4.00x^2+1.00x^1+6.00x^0
 At order 3 the Root is 3.000000

The 2 order polynomial is:     1.00x^2+-1.00x^1+-2.00x^0
 At order 2 the Root is 2.000000

The 1 order polynomial is:     1.00x^1+1.00x^0
 At order 1 the Root is -1.000000

There are 3 Roots for the given polynomial
```