CSE2202: Numerical Methods Lab
Online: 3 Section: A1
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: e: Determine the all possible real roots of the equation: $f(x) = x^3 - 7x^2 + 15x - 9 = 0$ using Modified Bisection Method.

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 $f(x) = x^3 - 7x^2 + 15x - 9 = 0$.
2. Your program must find the possible interval for equation where f1*f2<0 condition is satisfied.
3. Use Horner's rule to perform all iterations of the Modified Bisection Method until the estimated error $€_a$ falls below a level of $€_s = 0.0001$
4. Use appropriate functions from math header file.
5. 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
Possible interval -4.358899 and 4.358899
Number of Root   Approximate Root        Number of Iteration      Relative Error
1                 1.000004                53                      0.000006
2                 2.999637                73                      0.000008
3                 3.000370                74                      0.000008
```

Set B

Problem Statement: Determine the **multiple** real roots of the equation: $f(x) = x^5 - 5x^2 - 35x^3 + 125x^2 + 194x - 280 = 0$ using Newton's Method.

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$f(x) = x^5 - 5x^2 - 35x^3 + 125x^2 + 194x - 280 = 0$.
2. Write a function MaxRoot() that will return maximum possible root for the polynomial and consider this as your initial guess.
3. Use Horner's rule to perform all iterations of the Newton's Method until the relative estimated error $€_a$ falls below a level of $€_s = 0.001$
4. Use synthetic division to deflate the polynomial at lower degree. Write a function polynomialDeflation() that will return the coefficients of deflated polynomial.

   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 $n - 1$
5. Use appropriate functions from math header file.
6. Print the degree of the polynomial, roots found at each degree, after which iteration and relative error on that iteration.
7. Evaluate your approximate root using Horner's method if it return zero than print your root is closed to exact root. At the end print total number of roots you have found.

Sample Input/output:

```
 Enter values of coefficients:
 Coefficient x[5] = 1

 Coefficient x[4] = -5

 Coefficient x[3] = -35

 Coefficient x[2] = 125

 Coefficient x[1] = 194

 Coefficient x[0] = -280
Largest Possible root is 5.000000

 At order 5 the Root is 4.000000 after 3 iteration and relative error 0.000004
The Root is close to real Root

 At order 4 the Root is 1.000000 after 8 iteration and relative error 0.000063
The Root is not real Root

 At order 3 the Root is -2.000000 after 10 iteration and relative error 0.000000
The Root is close to real Root

 At order 2 the Root is -5.000000 after 14 iteration and relative error 0.000366
The Root is not real Root

 At order 1 the Root is 7.000000 after 16 iteration and relative error 0.000000
The Root is close to real Root
There are 5 Roots for the given polynomial
```