CMPT129: Assignment 1

PROBLEM 1: (100 points)

Develop a Test Plan for submission and an algorithm for your own use based on the information in this document. Then translate the algorithm into a C++ program. Your program will print a table with **numRows** rows to the screen and print and output file containing **numRows** tables, one table corresponding to each row in the table printed to the screen. Each row will show information about the approximations to the roots of a polynomial, P(X), of order up to 4 for a different initial range **intervalEndMin<=X<=intervalEndMax**.

The program will complete each of the following tasks

- The coefficients (**coef4, coef3, coef2, coef1, coef0)** of the polynomial P(X) are input by the user and verified by the program in the order listed. One coefficient must be verified before the next coefficient is read. These coefficients are used to evaluate the polynomial within your program. All coefficients must be in the range (-50.0 <= coef☺ <= 50.0) ☺ may be 0,1,2,3 or 4.
- Then the values of **intervalEndMin, IntervalEndMax** are supplied by the user and verified by the program. These values are the endpoints of an initial interval that may contain a root Y (P(Y) = 0), **intervalEndMin <= Y <= intervalEndMax.** The interval is used in the bisection algorithm used to determine the value of a root if one is found.
- Then the values of **numRows (1 <= numRows <= 25)** and **intervalShift (0.0 < intervalShift <= 20.0)** are input by the user and verified by the program. The value of **numRows** determines the number of rows printed in the table output to the screen and the number of tables printed to the output file. For the first row of the table printed to the screen
    - **intervalEndMin <= X <= intervalEndMax**.

  For the second row of the table printed to the screen .
    - **(intervalEndMin + intervalShift) <= X <= (intervalEndMax + intervalShift)**.

  For the N^{th} row of the table printed to the screen
    - **(intervalEndMin + (N-1)*intervalShift) <= X**
      <= intervalEndMax + (N-1)*intervalShift).
- The final input value to be input by the user and verified by the program is **nameFileOut** which contains the name of the output file.
- For each row of the table printed to the screen
  a) The first three columns of the table printed to the screen in are **intervalEndMin, intervalEndMax**, and **midpoint**. Each of these values should be printed as fixed point numbers with 5 digits after the decimal point.
  b) The next column is **P(midpoint)** for the final iteration
  c) The final column is the number of iterations needed to converge to an answer.

  Each of values in a) and b) above is printed in a field 13 characters wide
  The value in e) is printed in a field 5 characters wide.
- For each row of the table printed to the screen there will be one table printed to the output file. The table in the output file will show i**ntervalEndMin, P(intervalEndMin), midpoint, P(midpoint), intervalEndMax, P(intervalEndMax)**. There will be one column for each of these 6 quantities.

There will be one row for each iteration. This will be printed immediately before the step where the values of intervalEndMin or intervalEndMax are updated.

- Sample outputs are given below (and in files provided on Canvas) to specify the required formats and column titles for both the output to the screen and the output to the output file.

Outputs to screen and output file will be expected to match character for character with the provided output files.
Do not use arrays or vectors.
Use double variables not float variables.
Functions are not required but may be used if you wish.
Your code must comply with the class coding standard

How to verify input values

The value of each variable should be checked to assure that it is in the range immediately after it is read (before reading the next variable). If the variable being read is not in range then then a re-prompt for that variable should be printed to the screen and the value of the variable should be reread. If a total of four tries does not provide a value for the variable that is in range the program should print an error message then terminate.

All coefficients must be in the range (-50.0 <= coef☺ <= 50.0) ☺ may be 0,1,2,3 or 4. The prompt, re-prompt and error message used for the fourth order coefficient should be exactly as follows:

- `Please enter the value of the coefficient for the order 4 term`
  `(-50.0 <= coefficient <= 50.0):`
- `ERROR: The value of the order 4 term is out of range`
  `Please re-enter the value of the coefficient for the order 4 term`
  `(-50.0 <= coefficient <= 50.0):`
- `ERROR: The value of the order 4 coefficient is out of range`
  `Maximum number of tries entering data exceeded. Program terminating.`

For the messages for coefficients of orders 0, 1, 2, and 3 the messages are the same except the 4 in the messages above is replaced with 0, 1, 2, or 3 respectively.

Immediately after reading the values of the coefficients read **intervalEndMin** and checking that **intervalEndMin** is in range. read and verify **IntervalEndMax**. These two values are the **X** values at which the initial interval for the bisection method begins and ends respectively.
For **intervalEndMin** use the prompt and re-prompt messages below:

- `Please enter the minimum X value in the initial range`
  `(-60.0 <= minimum X value in the initial range <= 60.0):`
- `ERROR: The minimum X value in the initial range is out of range`
  `Please re-enter the minimum X value in the initial range`
  `(-60.0 <= minimum X value in the initial range <= 60.0):`

If a fourth incorrect value is input then print the following error message then terminate the program.

- `ERROR: minimum X value in the initial range is out of range`
  `Maximum number of tries entering data exceeded. Program terminating.`

Then read i**ntervalShift** and check that it is in range.

- **Please enter the maximum X value in the initial range**
  (-60.0 <= maximum X value in initial range <= 60.0):
- **ERROR: The maximum X value in the initial range is out of range**
  Please re-enter the maximum X value in the initial range
  (-60.0 <= maximum X value in the initial range <= 60.0):

If a fourth incorrect value is input then print the following error message then terminate the program.

- **ERROR: maximum X value in the initial range is out of range**
  Maximum number of tries entering data exceeded. Program terminating.

Immediately after reading and verifying the value of **intervalEndMax** read and verify the remaining values that control the size and content of the tables. Read and verify, in order **numRows** and **intervalShift**. First read and verify the number of rows in the table printed to the screen **(numRows).** Then read and verify **intervalShift**

- **Please enter the number of rows in the table**
  (1 <= number of rows in the table <= 25):
- **ERROR: The number of rows in the table is out of range**
  Please re-enter the number of rows in the table
  (1 <= number of rows in the table <= 25):

If a fourth incorrect value is input then print the following error message then terminate the program.

- **ERROR: the number of rows in the table is out of range**
  Maximum number of tries entering data exceeded. Program terminating.

Read i**ntervalShift** and check that it is in range. The amount that the initial interval for the bisection method is shifted between successive rows is intervalShift. For the Nth row of the table the initial interval for bisection will be

(intervalEndMin + (N-1)*intervalShift) <= X
<= intervalEndMax + (N-1)*intervalShift)

- **Please enter the shift in the starting interval between rows**
  (0.0 <= shift in the starting interval between rows <= 20.0):
- **ERROR: The shift in the starting interval between rows is out of range**
  Please re-enter the shift in the starting interval between rows
  (0.0 <= shift in the starting interval between rows <= 20.0):

If a fourth incorrect value is input then print the following error message then terminate the program.

- **ERROR: The shift in the starting interval between rows is out of range**
  Maximum number of tries entering data exceeded. Program terminating.

How to Evaluate Your Polynomial (you MUST use Horner's algorithm as described below)

Any time you wish evaluate a polynomial you must use the following steps to evaluate the polynomial. (Horner's algorithm).

- multiply **X** by **coef4** and add **coef3**
- multiply the sum from the line above by **X**
- add **coef2**
- multiply the sum from the line above by **X**
- add **coef1**
- multiply the sum from the line above by **X**
- add **coef0** (the constant coefficien

Horner's algorithm can be easily generalized for a polynomial of any order. As an equation for a 4^{th} order polynomial Horner's algorithm looks like

((((coef4*X + coef3)*X+coef2)*X+coef1)*X+coef0)

This approach requires many less multiplications that using the power function and is thus more efficient and accurate. **Note:** If you wish to have a polynomial of order less than 4 you can do so by setting the higher order coefficients to 0. For example for a quadratic equation (order 2) you could set both **coef4** and **coef3** to 0.

How to Determine the Root of a Polynomial of Order up to 4 Using the Bisection Method

Consider a polynomial P(X) of order up to 4. To find a root of this P(X) polynomial (a value of X for which P(X) = 0) you must use the bisection method in your program. Variables **IntervalEndMin** and **IntervalEndMax** are the ends of the interval in which you are searching for a root of the polynomial. **-60<intervalEndMin<60 and -60 <intervalEndMax<60**

You can used the bisection method for finding the root of a polynomial. The bisection method consists of the following steps

1. Check to see if either of the endpoints are roots
2. Find the length of the interval, **intervalLen = intervalEndMax – intervalEndMin;**
3. Find the midpoint of the interval, **midpoint = intervalEndMin + intervalLen/2.0**
4. Print the interval end values and the midpoint to the first row of the table on the screen.
5. Divide the interval into two equal subintervals **intervalEndMin to midpoint** and **midpoint to intervalEndMax**
6. Determine which half (if either) of the interval contains a root.
    a) If the subinterval contains an odd number of roots (1 or more) then the value of the polynomial at the ends of the subinterval will have different signs.
    b) If the subinterval contains no roots (or an even number of roots) the values of the polynomial at the ends of the subinterval will have the same sign.
    c) When implementing your bisection algorithm assume that if the value of the polynomial has the same sign at the two ends of the interval then there is no root in the interval.
7. Keep only one half of the interval which contains a root. If you retain the end of the interval including **intervalEnd1** then **intervalEnd2 = midpoint**. If you retain the end of the interval including **intervalEnd2** the **intervalEnd1=midpoint.**

a) If both halves of the interval contain roots choose the half of the interval containing **intervalEnd1**

b) If neither half contains an odd number of roots print the error message:
ERROR: No roots found
Then proceed to the next row

8. Repeat steps 1 to 5 until after step 4 either
   a) The value of the polynomial at the midpoint of the remaining interval is 0 (this means that **abs(P(midpoint)) < EPS**, where **EPS** is some small value below which the value is considered to be 0).
   b) The length of the interval is **<EPS**
   c) Use **EPS = 0.000001.**
9. If the bisection does not converge in <= 50 iterations print the error message
   ERROR: DID NOT CONVERGE
   To both the screen and the output file and proceed to the calculations for the next row of the table.

Addition points to keep in mind include:

- For each iteration (each time through steps 2 to 4), be sure that the endpoints of the interval/subinterval have been checked to see if any of them are roots!!!
- You are looking for any root. Your function needs to find only one root. To find the other roots of the polynomial you would use the function again, applying bisection to the same function on a different interval.

Be sure your code produces the exact output requested. All output should match the given examples in the test plan exactly (character for character). Details of formatting are given in the examples in the test plan and must be matched character by character for full points. No extra input or output that is not shown in the test plan should be added.

Be sure to write your code so that the style of your code is consistent with the following parts of the course coding standard:

- Declare and initialize all variables at the start of the main function (Before any code that executes).
- Declare and initialize one variable per line of code.
- Do not do calculations in the same statement as the declaration
- Use descriptive variable names so your code is self-documenting.
- Use named constants for all your numerical constants. Create and initialize all named constants at the start of the main function
- Begin each program with a block of comments that give the author or authors of the code, the date the code was completed, and a very brief description of what the program does. Place comments in your code to add information not clear from your self documenting code.
- When using if statements or loops , always use blocks that is always use { }, always put the { and the close } on their own line, and always indent code that is contained within a block (within the { })
- Indent for each addition block delineated by { }

The following examples provide a test plan and a model for the format of the desired output. The text, spacing, and formatting shown in the examples must match your results exactly (character by character). These examples do not test all possible outcomes of the program. When the program is graded any possible outcome may be tested for. To determine exact spacings please refer to the supplied output files not the examples below. The numbers shown in red are entered by the user running the program

1) Normal case 1

```
Please enter the value of the coefficient for the order 4 term
(-50.0 <= coefficient <= 50.0): 0
Please enter the value of the coefficient for the order 3 term
(-50.0 <= coefficient <= 50.0): 4.0
Please enter the value of the coefficient for the order 2 term
(-50.0 <= coefficient <= 50.0): -18
Please enter the value of the coefficient for the order 1 term
(-50.0 <= coefficient <= 50.0): 6.0
Please enter the value of the coefficient for the order 0 term
(-50.0 <= coefficient <= 50.0): 8
Please enter the minimum X value in the initial range
(-60.0 <= minimum X value in the initial range <= 60.0): -19.0
Please enter the maximum X value in the initial range
( -60.0 <= maximum X value in the initial range <= 60.0): 21.0
Please enter the number of rows in the table
(1 <= number of rows in the table <= 25): 16
Please enter the shift of the starting interval between rows
(0.0 <= shift of starting interval between rows <= 20.0): 2.0
Enter the name of the output file outCase1.txt
    intEndMin    intEndMax     midpoint         root    N
   -19.000000    21.000000     1.000000     1.000000    1
   -17.000000    23.000000     3.000000     4.000000   27
   -15.000000    25.000000     5.000000    -0.500000   27
   -13.000000    27.000000     7.000000     4.000000   27
   -11.000000    29.000000     9.000000     4.000000    3
    -9.000000    31.000000    11.000000     1.000000    2
    -7.000000    33.000000    13.000000     4.000000   27
    -5.000000    35.000000    15.000000    -0.500000   27
    -3.000000    37.000000    17.000000     4.000000   27
    -1.000000    39.000000    19.000000     4.000000    3
     1.000000    41.000000    21.000000     1.000000    1
     3.000000    43.000000    23.000000     4.000000   27
     5.000000    45.000000    25.000000------------ERROR:  no roots in the interval
     7.000000    47.000000    27.000000------------ERROR:  no roots in the interval
     9.000000    49.000000    29.000000------------ERROR:  no roots in the interval
    11.000000    51.000000    31.000000------------ERROR:  no roots in the interval
```

Output file is available in Canvas, too long to include here with the console output

2) Normal case 2 (with maximum number of iterations set to 24

```
Please enter the value of the coefficient for the order 4 term
(-50.0 <= coefficient <= 50.0): 0
Please enter the value of the coefficient for the order 3 term
(-50.0 <= coefficient <= 50.0): 4
Please enter the value of the coefficient for the order 2 term
(-50.0 <= coefficient <= 50.0): -18.0
Please enter the value of the coefficient for the order 1 term
(-50.0 <= coefficient <= 50.0): 6
Please enter the value of the coefficient for the order 0 term
(-50.0 <= coefficient <= 50.0): 8.0
Please enter the minimum X value in the initial range
(-60.0 <= minimum X value in the initial range <= 60.0): -9
Please enter the maximum X value in the initial range
( -60.0 <= maximum X value in the initial range <= 60.0): 11
Please enter the number of rows in the table
(1 <= number of rows in the table <= 25): 8
Please enter the shift of the starting interval between rows
(0.0 <= shift of starting interval between rows <= 20.0): 4.0
Enter the name of the output file outCase2.txt
    intEndMin    intEndMax     midpoint         root    N
    -9.000000    11.000000     1.000000     1.000000    1
    -5.000000    15.000000     5.000000-------------ERROR: did not converge
    -1.000000    19.000000     9.000000     4.000000    2
     3.000000    23.000000    13.000000-------------ERROR: did not converge
     7.000000    27.000000    17.000000-------------ERROR:  no roots in the interval
    11.000000    31.000000    21.000000-------------ERROR:  no roots in the interval
    15.000000    35.000000    25.000000-------------ERROR:  no roots in the interval
    19.000000    39.000000    29.000000-------------ERROR:  no roots in the interval
```

Output file is available in Canvas, too long to include here with the console output

3) Out of range values for the coefficient of order 4
   Too many tries entering the initial estimate

```
Please enter the value of the coefficient for the order 4 term
(-50.0 <= coefficient <= 50.0): 55
ERROR: The value of the order 4 term is out of range
Please re-enter the value of the coefficient for the order 4 term
(-50.0 <= coefficient <= 50.0): -55.55
ERROR: The value of the order 4 term is out of range
Please re-enter the value of the coefficient for the order 4 term
(-50.0 <= coefficient <= 50.0): 743.4
ERROR: The value of the order 4 term is out of range
Please re-enter the value of the coefficient for the order 4 term
(-50.0 <= coefficient <= 50.0): -333.3
ERROR: The value of the order 4 coefficient is out of range
Maximum number of tries entering data exceeded. Program terminating.
```

No Output file should be created by this example