Lab2 Assignment

Evaluation of Polynomial

Algorithms

In accordance with https://en.wikipedia.org/wiki/Polynomial - In mathematics, a *polynomial* is an expression consisting of variables (also called indeterminates) and coefficients, that involves only the operations of addition, subtraction, multiplication, and non-negative integer exponents of variables.

A polynomial can be written as (where x is a variable, a_i are constants, and N>=0)

$$p(x) = a_N x^N + a_{N-1}x^{N-1} + \cdots + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

Suppose you have few polynomials, and you and you need to perform evaluate operation on them. Write a program to solve this polynomial problem. Also estimate evaluate operation efficiency - how much CPU time is used, how many times scalar arithmetic operators +, - are called, and how many times scalar arithmetic operator * is called. Write 3 algorithms to implement evaluation of the polynomial.

1. Calculate $p(x) = a_N x^N + a_{N-1}x^{N-1} + \cdots + a_3 x^3 + a_2 x^2 + a_1 x + a_0$
2. Calculate $p(x) = x, x^2 .. x^{N-1}, x^N$
   And store each term in a vector
3. Calculate using Horner's rule (see https://en.wikipedia.org/wiki/Horner%27s_method for details). In summary, any polynomial can be rewritten as
   $p(x) = a_0 + x\left(a_1 + x\left(a_2 + x(a_3 + \cdots + x(a_{N-1} + xa_N)\ldots)\right)\right)$
   The following pseudocode can be used to evaluate the polynomial (where **x** is a given)
   p = 0; for ($i = N$; i >= 0; --i) p = **x** * p + a_i ;

Program Algorithm

Use container vector in C++ and Java and array in Python

Assume, that we will rewrite $p(x) = a_N x^N + a_{N-1}x^{N-1} + \cdots + a_3 x^3 + a_2 x^2 + a_1 x + a_0$
as $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{N-1}x^{N-1} + a_N x^N$
The program shall do the following

1. read from the input file
   - *value to evaluate (double)x*
   - value of coefficients $a_0\ a_1\ a_2 ..\ a_{N-1}\ a_N$ separated by spaces
2. apply evaluation of polynomial: 3 algorithms
3. estimate how much CPU time each *operation evaluation* took, print it to the output file
4. count how many times scalar arithmetic operators +, -, and * were called, print it to the output file separately.
5. print the operation result to the output file

Note that size of the vector(C++ and Java) will be *N+1 (array in Python)*

For example, for evaluation value $x = 2$ and polynomial

$p(x) = 6 + 8x + 10x^2 + 0x^3 + 7x^4 + 5x^5$ and you need to calculate $p(2)$

the input file contains

2

6 8 10 0 7 5

the output file will contain

```
2
6 8 10 0 7 5
Algortihm1: => p(2)= 334, 1.6128e-05 nanoseconds,
5 +-operations, 5 *operations, 5 pow function calls
Algortihm2: => p(2)= 334, 4.399e-06 nanoseconds,
5 +-operations, 5 *operations, 5 pow function calls
Algortihm3: => p(2)= 334, 0 nanoseconds,
6 +-operations, 6 *operations
```

Driver

The main program function (the driver) has to implement the program algorithm as described previously in this document.

The driver has to read data from an input file and print results to an output file. For the format of the input and output files, see example above.

Submit Driver program, in.txt, out.txt r5.txt and output of your program when input is r5.txt file.