CS 200 – Intro to Programming

Assignment 3 [Fall 2018]

Release Date: Saturday 19th October 2018, 12:00 AM

Due Date: Sunday 28th October 2018, 11:55 PM

Please keep in mind the following guidelines:

- Do not share your program code with anyone.
- Do not copy code from the internet.
- If you receive any assistance, mention the part of code in which you received assistance.
- You must be able to explain any part of your submitted code.
- All submissions are subject to automated plagiarism detection.

Submission:

You have to submit all the .cpp files containing source code. Zip all .cpp files into one file named as <your8DigitRollNumber> .zip and submit the zip file.

A sheet named “Asgn_3_Queries” has already been made on the seating plan link which we’ll keep updated with your latest questions. This sheet should be the first thing you refer when you have a question, email one of the TAs of your question is not answered there.

Please note that you have to work in separate header and cpp files when defining your class. You must submit a single .zip file which would include 3 files: myPolynomial.h, myPolynomial.cpp and yourRollNumber_Task1_Asgn3.cpp

Task 1: (70 marks)

Implement the polynomial class:

```
class myPolynomial {
        private:
                int terms;
                float *degreeArray;
                float *coeffArray;
};
```

Consider a polynomial $4x^6-2x^3+6x^2+1$ which would be stored in an object of our class as:

terms = 4 **degreeArray = [6, 3, 2, 0]** **coeffArray = [4, -2, 6, 1]**

The format of storage is also given in the sample file “samplePolynomial.txt”

Make sure that the degree array is sorted in descending order so that the highest power term appears first. Also, there should be no duplicates in the degree array and it should only be allocated according to the total terms present in the polynomial.

You must implement the following in myPolynomial class:

a. constructor, destructor, copy constructor

b. + operator for addition of two polynomials

c. - operator for subtraction of two polynomials

d. * operator for multiplication of two polynomials

e. = assignment operator

f. == comparison operator

g. printPolynomial() // should print a polynomial in $4x^6-2x^3+6x^2+1$ form

h. readFromFile()

i. … any other helper functions that you might find useful

Example Menu:

0. Exit
1. Input polynomial a from file (for this option enter filename from user)
2. Input polynomial b from file (for this option enter filename from user)
3. Add a and b to give c
4. Multiply a and b to give c
5. Subtract a from b to give c
6. Print a
7. Print b
8. Print c
9. Check a == b
10. … … … … and so on

Example run:

Following code should work if called from a caller function:

```
myPolynomial a, b, c;
a.readFromFile(“samplePolynomialA.txt”);
b.readFromFile(“samplePolynomialB.txt”);
c = a + b;
a = c - b;
b = a * b;
a.printPolynomial();
```

```
b.printPolynomial();
if(a==b) cout << “same polynomial”;
… … …
```

Note:

The arrays should be allocated according to the total terms of the polynomial. There should be no memory leaks. You have to find out how addition, multiplication and subtraction of polynomials takes place. Also, you have to make sure that the resulting polynomial degrees are sorted in descending order without any duplicates. You can implement any required helper private function.

Task 2: (30 marks)

Binomial coefficients are the numeric factors of the products in a power of a binomial such as $(x + y)^n$. For example, $(x + y)^2 = x^2 + 2\,x\,y + y^2$ has the coefficients 1 2 1. Binomial coefficients can be calculated using Pascal's triangle(shown below).

```
                1                       n = 0

            1       1

        1       2       1

     1      3       3       1

1       4       6       4       1       n = 4
```

Each new level of the triangle has 1's on the ends; the interior numbers are the sums of the two numbers above them. Write a program that includes a **recursive** function to produce a list of binomial coefficients for the power n using the Pascal's triangle technique. For example,

```
Input = 2          Output =  1   2   1
Input = 4          Output = 1   4   6   4   1
```

Task 3: (15 marks bonus)

Write a **recursive** function that accepts two integer arguments into the parameters x and y. The function should return the value of x times y. You are required to compute the product as repeated addition, for example: 7*4=4+4+4+4+4+4+4. Write a program that reads two numbers from the user and uses this function to find the product of the given numbers. You may assume that the numbers entered by the user are both greater than zero.