Assignment 3
Object Oriented Programing

Topic: operator overloading

Total Marks: 100

Due Date: 20st December 2017 11:59:59pm

Instructions

1. Attempt all questions.
2. Copy/Cheating is **STRICTLY PROHIBITED.** If anyone caught doing cheating simply whole assignment will be canceled.
3. You can discuss assignment questions with other students but write your OWN code. Don't share your code with anyone.
4. Late submission will cause 50% deduction of marks.
5. Email Address for submission: bcsf16oop@gmail.com
6. Best of luck ☺

Class Matrix {

Private:

Int **arr, rows,cols;

Public:

1. Write default constructer. In default constructer pointer should be null and rows and columns should be 0.
2. Write parameterized constructer.
3. Write copy constructer.
4. Write destructor.
5. Write all getter and setter.
6. Overload stream operators (cin and cout).
7. Overload mathematical operators (+,-,*,/). Division can be performed with a number and with a matrix as well. For example if user gives matrix $\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$ and a number 2 then you need to divide each element by that number. In this case the result will be $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. In case of divide by matrix divide corresponding elements of both matrixes.
   $\begin{bmatrix} 10 & 20 \\ 100 & 200 \end{bmatrix} / \begin{bmatrix} 5 & 10 \\ 50 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 100 \end{bmatrix}$
8. Overload all six relational operators (==, !=, <, <=, >, >=). If sum of all elements of matrix A is less than sum of all matrix of B then return true otherwise false.

$A = \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix} < B = \begin{bmatrix} 11 & 12 \\ 13 & 14 \end{bmatrix}$ = True

$A = \begin{bmatrix} 11 & 12 \\ 13 & 14 \end{bmatrix} < B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ = False

9. Overload assignment operator.
10. Overload both (++,--) prefix and postfix operators. ++ means adding 1 to all elements of matrix and – means subtracting 1 from all elements of matrix.
11. Write function that take transpose of the matrix.
12. Implement {int, double, Set} conversion operator so that a Matrix can be converted to integer, double and Set (You implemented Set class in assignment 2. So you need set class here.)
    a. Converting a matrix into int means adding all its element and return the sum.
    b. Converting a matrix into double means adding all its element and return the sum.
    c. Converting a matrix into Set means converting 2-D array into 1-D array.

};

Int main () {

Main should give user following menu.

- Perform Mathematical Operation.
  - Add
  - Subtract
  - Multiply
  - Divide
  - Go Back
- Perform Relational Operation.
  - Equal
  - Not Equal
  - Less Than
  - Less Than and Equal
  - Greater Than
  - Greater Than and Equal
  - Go Back
- Perform Unary Operation.
  - Prefix ++
  - Prefix --
  - Postfix ++
  - Postfix --
  - Transpose
  - Go Back
- Sorting (This is optional and bonus task. This will take ***array of matrixes*** from user and sort them and print them.)
  - Ascending
  - Descending

- Conversion
  - Convert Matrix to Int.
  - Convert Matrix to double.
  - Convert Matrix to Set
  - Go Back.
- Exit

}

Note: Regarding prefix and postfix operator be careful! If user selects *Prefix ++* then you need to take two matrixes from user and do **a = ++b** and display result of **a** and **b** on screen and if user selects *Postfix ++* then you need to take two matrixes from user and do **a = b++** and display the contents of both matrix to the user. Look at the following example for more clarity.

A = $\begin{bmatrix} 7 & 56 \\ 6 & 54 \end{bmatrix}$ B = $\begin{bmatrix} 4 & 6 \\ 60 & 45 \end{bmatrix}$

Performing A = ++B

A = $\begin{bmatrix} 5 & 7 \\ 61 & 46 \end{bmatrix}$ B = $\begin{bmatrix} 5 & 7 \\ 61 & 46 \end{bmatrix}$

Performing A = B++

A = $\begin{bmatrix} 4 & 6 \\ 60 & 45 \end{bmatrix}$ B = $\begin{bmatrix} 5 & 7 \\ 61 & 46 \end{bmatrix}$

Do same for the unary operator --