**Instructions:**

1. C++ codes must be submitted on LMS before: ????
2. Submit a single zipped file containing two folders (one for each question)
3. Write your name, section and registration number before the start of each code (in comments).
4. You already know the consequences of cheating.
5. Whether mentioned or not, provide copy constructor, destructors, getters and setters, display appropriate input and output messages and make a complete main program too.

**Question 1:** *(****HugeInteger*** *Class)* Create a class HugeInteger that uses a dynamically created 40-  
element array of digits to store integers as large as 40 digits each. Use operator overloading to  
provide member functions for input, output, add and subtract. For comparing HugeInteger objects,  
provide overloaded operators which perform the functions: isEqualTo, isNotEqualTo,  
isGreaterThan, isLessThan, isGreaterThanOrEqualTo and isLessThanOrEqualTo—each of these  
is a “predicate” function (a function that simply returns true if the relationship holds between the  
two HugeIntegers and returns false if the relationship does not hold). Also, provide a predicate  
function isZero. Use operator overloading to provide member functions multiply, modulus and divide too.

**Question 2:** *(****TicTacToe*** *Class)* Create a class TicTacToe that will enable you to write a  
complete program to play the game of tic-tac-toe. The class contains as private data a 3-by-3  
two-dimensional array of integers. The constructor should initialize the empty board to all zeros.  
Allow two human players. Wherever the first player moves, place a 1 in the specified square.  
Place a 2 wherever the second player moves. Each move must be to an empty square. After each  
move, determine whether the game has been won or is a draw. If you feel ambitious, modify  
your program so that the computer makes the moves for one of the players. Also, allow the  
player to specify whether he or she wants to go first or second.

**Question no: 3** Create your own CustomString class with following functions:

CustomString();//default constructor

~CustomString();

CustomString(char \*ptr);

CustomString(const CustomString &cc);

void setCustomString(char \*ptr);

const char\* getCustomString();

bool isEqualTo(CustomString c);

int length();//return size of CustomString

bool isEmpty();

//return last index of given character if exist in CustomString otherwise return -1

int lastIndexOf(char ch);

//replace all occurrence of oldChar with newChar

CustomString replace(char oldChar, char newChar);