# Data Structures & Algorithms for Problem Solving
## Assignment 1
### Deadline: 10ᵗʰ September 2020 , 11.55 pm

## IMPORTANT POINTS

1. Languages Allowed : C/C++
2. Submission Format: Roll No_Question No.cpp Ex: For question 1, 2019201001_Q1.cpp.Copy all the codes in a folder with name as your roll no. and submit the zip file in moodle. Ex: 2019201001_A1.zip
   Note: **All those submissions which are not in the specified format or submitted after the deadline will be awarded 0 in assignment.**
3. **C++ STL are not allowed in any of the questions.**

## Any case of plagiarism will lead to 0 in assignment or "F" in the course.

## Problem 1:

**Statement:** You have to design a generic calculator that takes as input an expression and evaluates as outputs the answer.
**Use self implemented generic stack. No STL allowed.**

**Terminology**:  Generic stack= A single expression can contain both int,double and the answer should be accordingly.

Example: 1. Input : (4+5)/6*3          Output : 4.5

        2. Input: (4+5.6)/3.2+(4+3.888)   Output: 10.888

**Operations to be implemented:** Addition, Subtraction, Multiplication, Division and Modulo

**Clarifications :** The input given will be a correct mathematical statement with matching brackets(if any). You need not worry about Negative numbers as we haven't mentioned unary – operator.

**Evaluation parameters**: Output correctness, Accuracy of operations and performance.

## Problem 2:

**Statement**: Your task is to create a large integer library, similar to what we have in Java as BigInteger. Your library should provide functionalities to store arbitrarily large integer and perform basic math operations.

**Operations to be implemented**: Fast exponentiation, GCD of 2 integers and factorial.

**Evaluation parameters**: Accuracy of operations and performance.

## Problem 3:

**Statement**: Implementation of deque.

What is deque?

- Deque is the same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container.

- They support insertion and Deletion from both ends in amortized constant time.
- Inserting and erasing in the middle is linear in time.

What is expected as solution?

The C++ standard specifies that a legal (i.e., standard-conforming) **implementation of deque must satisfy the following performance requirements:**

- deque() - initialize a blank deque.
- deque(n,x) - initialize a deque of length n with all values as x.
- push_back(x) - append data x at the end.
- pop_back() - erase data at the end.
- push_front(x) - append data x at the beginning.
- pop_front() - erase data at the beginning.
- front() - returns the first element(value) in the deque.
- back() - returns the last element(value) in the deque.
- empty() - returns true if deque is empty else returns false.
- size() - returns the current size of deque.
- **resize(x, d)** - changes the size dynamically. If the new size is greater than the current size of the deque, then fill the empty space with the default value d.
- clear() - remove all elements of deque.
- D[n] - returns the nth element of the deque.

**Evaluation parameters**: Accuracy of operations and performance.

**Note : For all the questions, accuracy will be tested on the basis of test cases passed which will be provided during evaluation.**