

University of New Brunswick
Faculty of Computer Science
CS2383: Data Structures and Algorithms
*Programming Homework Assignment 1, **Due Time, Date** 5:00 PM, October 26, 2018*

Student Name: _____ Matriculation Number: _____

Instructor: Rongxing Lu

The marking scheme is shown in the left margin and [100] constitutes full marks.

- [40] 1. Given an integer array $A = \{7, 4, 8, 2, 5, 3, 9\}$,
- [20] (a) Write your own Java source code named `SelectionSort.java` to implement Selection Sort algorithm on the array A . Please finish your code in the following template, where “XXSort” is replaced with “SelectionSort”.
- [20] (b) Write your own Java source code named `InsertSort.java` to implement Insertion Sort algorithm on the array A . Please also finish your code in the following template, where “XXSort” is replaced with “InsertSort”.

```
public class XXSort
{
    public static void main()
    {
        int[] A = {7, 4, 8, 2, 5, 3, 9};
        sort(A);
        show(A); // display the sorted result A
    }

    public static void sort(int[] A)
    {
        ...
    }

    public static void show(int[] A)
    {
        ...
    }

    ... // add all other functions you need
}
```

- [15] 2. Write a Java source code named “Stack.java” to implement the following functions of a Stack using an integer array:
- `stack_init()`: Make the stack empty
 - `empty()`: return true if the stack is empty. Return false if the stack is not empty

- push(val): add the item val to the stack
- pop(): remove the item most recently added to the stack
- top(): return the item most recently added to the stack, but do not remove it.

the program reads in integer values as input:

- for a positive integer input value greater than 0, the program will push the value into the stack
- for any negative integer input value, the program will return the most recently added value from the stack if it is non-empty, i.e. top
- for the input value 0, the program will remove the most recently added value from the stack if it is non-empty, i.e. pop

test your program with the following set of input:

Input	Expected Output
0	"Error - Stack is empty"
78	
456	
-1	456
0	
-1	78
60	
33	
-1	33
0	
-1	60
0	
0	
-1	"Error - Stack is empty"
0	"Error - Stack is empty"

[15] 3. Write a Java source code named "Queue.java" to implement the following functions of a Queue using an integer array:

- queue_init(): Make the queue empty
- empty(): return true if the queue is empty. Return otherwise
- enqueue(val): add the item val to the queue
- dequeue(): remove the item least recently added to the queue
- front(): return the item least recently added to the queue, but do not remove it.

the program reads in integer values as input:

- for a positive integer input value greater than 0, the program will insert the value into the queue, i.e. enqueue
- for any negative integer input value, the program will return the least recently added value from the queue if it is non-empty, i.e. front
- for the input value 0, the program will remove the least recently added value from the queue if it is non-empty, i.e. dequeue

test your program with the following set of input:

Input	Expected Output
0	"Error - Queue is empty"
78	
456	
-1	78
0	
-1	456
60	
33	
0	
-1	60
0	
-1	33
0	
-1	"Error - Queue is empty"
0	"Error - Queue is empty"

[20] 4. A bracket sequence consisting of '(' and ')' is defined to be valid as follows:

- An empty sequence is valid.
- If X is a valid bracket sequence, then (X) is a valid bracket sequence.
- If X and Y are valid bracket sequences, then the concatenation of X and Y, Z=XY, is a valid bracket sequence

For example, "()", "(())", and "(()())" are all valid bracket sequences, while "(" and "())" are invalid bracket sequences. Given a bracket sequence, write an algorithm named "CheckBrackets.java" to judge whether it is valid or not.

[10] 5. Monkey & peaches problem. A monkey picked a pile of peaches on Monday. From that Monday to Saturday, no matter how many peaches the monkey had, each day the monkey ate $\frac{1}{2}$ of its peaches in the morning and ate one more peach in the afternoon. (For example, at the end of Wednesday, the monkey had n peaches left, then the monkey would eat $\frac{n}{2} + 1$ peaches on Thursday.) Then, on Sunday morning, the monkey found only one peach left. Do you know how many peaches the monkey picked on Monday? The answer is 190. Now, please write a recursive algorithm named "MonkeyEatPeach.java" to compute the total number of peaches that monkey picked on that Monday.