LAB 05: Stack and Queue

CS211 – Data Structures and Algorithms
Usman Institute of Technology
Fall 2020

- **How to submit:**
  - Online: Submit on your respective MS Teams.

A. Create a class Stack and implement the Stack operations in the following order.

1. Add a constructor of the class that takes one argument size in order to set the size of the stack.

```
Class Stack:
    def __init__(self,size):
        // your code goes here
```

2. Add a function **IsEmpty()** that returns true if the stack is empty otherwise returns False.

```
    def IsEmpty(self):
        // your code goes here
```

3. Add a function **Push()** which takes an argument item to insert data into the stack. The function should also check the overflow condition.

```
    def Push(self, item):
        // your code goes here
```

4. Add a function **IsFull()** that returns true if the stack is full otherwise returns False.

```
    def IsFull(self):
        // your code goes here
```

5. Add a function **Pop()** that removes the elements from the stack. The function should also check the underflow condition.

```
    def Pop():
        // your code goes here
```

6. Add a function **Peek()** which returns the value from top of the stack.

```
    def Peek(self):
        // your code goes here
```

7. Add a function **Count()** that returns number of elements in the stack.

```
    def Count():
        // your code goes here
```

8. Add a function **Print()** which prints all the elements of the stack.

```
    def Print(self):
        // your code goes here
```

B. Create a class Queue and implement the Queue operations in the following order.

1. Add a constructor of the class that takes one argument size in order to set the size of the Queue.

```
Class Queue:
    def __init__(self,size):
        // your code goes here
```

2. Add a function **IsEmpty()** that returns true if the queue is empty otherwise returns False.

```
    def IsEmpty(self):
        // your code goes here
```

3. Add a function **Enqueue()** which takes an argument item to insert data into the queue. The function should also check the overflow condition.

```
    def Enqueue(self, item):
        // your code goes here
```

4. Add a function **IsFull()** that returns true if the queue is full otherwise returns False.

```
def IsFull(self):
    // your code goes here
```

5. Add a function **Dequeue()** that removes the elements from the queue. The function should also check the underflow condition.

```
def Dequeue(self):
    // your code goes here
```

6. Add a function **Count()** that returns number of elements in the queue.

```
def Count():
    // your code goes here
```

7. Add a function **Print()** that prints number of elements in the queue.

```
def Print():
    // your code goes here
```

C. Given an expression string exp , write a function StringExp() to examine whether the pairs and the orders of "{","}","(",")","[","]" are correct in exp.

```
def StringExp():
    // your code goes here
```

For example: the program should return true for exp = "[()]{}{[()()]()}" and false for exp = "[(])"