Assignment # 4: Stacks

1. Give one real-life example of how **stacks** can be used in real life. Then, describe or illustrate its concept and usage.

2. Given a class of Stack which is defined as follows:

```
class ArrayStack:
  def __init__(self):
    """ Create an empty stack. """
    self._data = []    # Initiate a nonpublic list instance

  def __len__(self):
    """ Return the number of elements in the stack. """
    return len(self._data)

  def is_empty(self):
    """ Return True if the stack is empty. """
    return len(self._data) == 0

  def push(self, element):
    """ Add an element to the top of the stack. """
      self._data.append(element)  # new item stored at end of
     list

  def top(self):
     """ Return (but do not remove) the element at the top of
     the stack. Raise an exception if the stack is empty. """
    if self.is_empty():
      print('Stack is empty')
      raise Empty('Stack is empty') # Calling subclass Empty
    return self._data[-1] # the last item in the list

  def pop(self):
      """ Remove and return the element from the top of the
     stack. Raise an exception if the stack is empty. """
    if self.is_empty():
      print('Stack is empty')
         raise Exception('Stack is empty') # Alternate way to
     call subclass Empty
    return self._data.pop() # remove last item from the list
```

Given a stack class as defined above, a method to create an empty stack is:

S = ArrayStack()

What values are returned during the following series of stack operations, if executed upon an initially empty stack?

Operation	Return Value	Stack values
S.push(9)		
S.push(5)		
S.top()		
S.pop()		
S.push(7)		
S.push(1)		
len(S)		
S.pop()		
S.pop()		
S.pop()		
S.pop()		
S.is_empty()		
S.push(4)		
len(S)		

3. Based on a Stack class as defined in question 2, write the pseudocodes or python codes for the following tasks.

   - Create 3 empty stacks: A, B and C.

   - Add new items including “Ant”, “Bird”, “Cat”, and “Dog”, respectively, into stack “A”.

   - Add new items including “Tony”, “Sara”, and “Adam”, respectively, into stack “B”.

- Transfer the “Dog” and “Cat” items from stack “A” to stack “B”

  [*Only use stack methods to move elements, list methods are not allowed.*]

- Add “Bear” into the stack “A”. Then sort the items in the stack “A” to be alphabetically ordered by ‘A’ to ‘Z’ where items starting with an ‘A’ are at the top of the stack and items starting with a ‘Z’ are at the bottom of the stack.

  [*Only use stack methods to move elements, list methods are not allowed.*]

  Hint: Use stack C to store elements that are not in the correct order.

- Remove all items in stack “B” by using a Python loop.

4. Based on a Stack class as defined in question 2, implement a **pop_all method** that removes all items in a stack using Python code or pseudocode or a flowchart.

   Hint: Use a loop.
   Hint 2: When you are implementing a method, you are allowed to use the List method.

```
class ArrayStack:
    .....
    (as defined in question 2)
    .....
    def pop_all(self):
        #Add your code here
```

5. Suppose an initially empty stack S has executed a total of 10 push operations, 5 top operations, and 9 pop operations, 4 of which raised Empty errors that were caught and ignored. What is the current size of S? Also, show your calculation of how you obtain the size.

6. [Hard] Write a pseudocode or a flowchart or a Python code to solve a balanced grouping symbols problem (e.g. (), {}, and []). Use a **stack approach** and not string manipulation approach.

```
def IsBalanced(str):

stack = ArrayStack()
....
....
#Add your code here
```