**Task 01 Answer.**

For the above requirement, I have chosen **Stack** as the most appropriate data structure.

In computing, stack is a data structure used to store a collection of objects. It is called a stack because it behaves like a real world stack.

**For example** - Piles of Balls.

A pile of books.

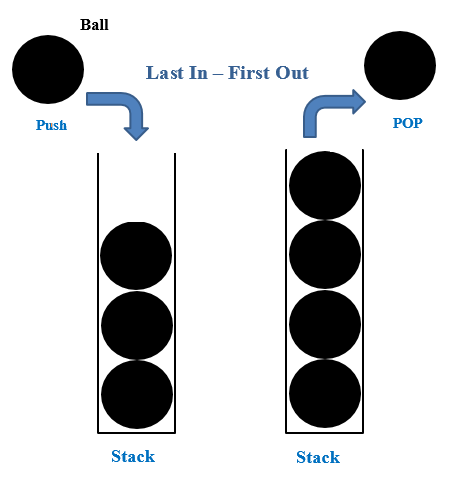
Piles of cards.

We can only place or remove the ball at the top of the stock. Similarly, the stack allows only one end of all data operations. We can only add to the stack top element at any time.

This feature makes it **LIFO** data structure. **LIFO** stands for Last-in-first-out. Here, the element which is placed (inserted or added) last, is accessed first. In stack terminology, insertion operation is called the **Push** operation and the removal operation is called the **POP** operation.

**Stack Representation.**

The diagram below shows the stack and its functionality.



**Figure 1 : Stack Representation**

**Basic Operations.**

Stack operations may include stack initialization, use, and disabling. Apart from these basics, stack is used for the following two primary operations.

1. **Push () - to stack an element.**

The **Push ()** method allows one or more elements to be added to the ends of an array. The **Push ()** method returns the value of the length attribute that specifies the number of elements in the array.

1. **POP () - Removing a stack element.**

Accessing content while being removed from stack is called a pop operation. When executing the **POP ()** array, the data element is not actually removed, but instead reduced to a lower point in the stack to point to the next. But when implementing linked lists, **POP ()** actually removes the data element and invalidates the memory space.

***In addition to using stack efficiently, we also need to check stack status. To do this, add the following functionality to the stack.***

1. **Peek ()** - this allows you to see what is the top element of the stack without leaving the element.

**Advantages and Disadvantages of Stack**

**Advantages.**

* Allows repetition.
* In stack we can allocate memory dynamically
* In the stack we can easily add or remove partitions.

**Disadvantages.**

* Dynamic memory allocation is a waste of memory space if we don't use all the memory space.