

Data Structures & Algorithms LAB

(BSCS-F18 Morning & Afternoon)

Lab # 4

Task # 1

You are given two arrays of integers, both containing exactly **n** integers. Write a function to determine the **intersection** (i.e. common elements) of these two arrays, as described below. Assume that there are **no duplicates** in either of the two arrays.

Assume that both the input arrays (**A** and **B**) are **sorted** in **increasing order**; implement the following function to determine the intersection of the two arrays:

int intersection3 (int* A, int* B, int* C, int n)

The worst-case time complexity of this function should be **$O(n)$** . You are NOT allowed to allocate any new array within this function.

Task # 2

Stack is a LIFO (Last in First out) structure or we can say FILO (First in Last out). **push()** function is used to **insert** new **elements** into the **Stack** and **pop()** function is used to remove an **element** from the **stack**. Both **insertion** and removal are allowed at only one end of **Stack** called **Top**.

Stack class is given below

```
Class Stack
{
    private :
        int *S;
        int maxsize, top;
    // constructor
    //destructor
    bool push(int val)
    bool pop(int &val)
    bool isfull()
    bool is empty()
}
```

Implement following function of stack.

- 1- // **constructor**
- 2- //**destructor**
- 3- **bool push(int val)**
- 4- **bool pop(int &val)**
- 5- **bool isfull()**
- 6- **bool is empty()**

Task # 3

Reverse a number using Stack

Let x= 78912453

And reverse = 35421987

Note : Assume number contains 9 digits at max. So maximum size of stack is 9.