

**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**  
**Computer Engineering Department**  
Program: MBA Tech. Sem III

**Course: Data Structure**  
**List of Experiments**

LAB Manual

PART A

(PART A : TO BE REFERRED BY STUDENTS)

**Experiment No.02**

**A.1 Aim:**

To study and implement concept of array and Stack data structure.

**A.2 Prerequisite:**

1. Knowledge of different operations performed on Stack data structure
2. Fundamental concepts of C/C++.

**A.3 Outcome:**

After successful completion of this experiment students will be able to

1. Identify the need of appropriate selection of data structure
2. Identify the steps of array and stack data structure selection.
3. Implement stack data structure to solve the given problem
4. Enlist the applications of stack data structure.

**A.4 Theory:**

**A.4.1. Introduction of Stack**

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last in first out) or FILO (First in last out).

The functions performed on stack are:

1. **Push:** Adds an item in the stack. If the stack is full, then it is said to be an overflow condition.

2. **Pop:** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an underflow condition.
3. **Peek:** Returns the top most element from the stack.

#### **Applications of Stack:**

1. Balancing of symbols
2. Infix to postfix\prefix conversion
3. Redo-undo features at many places like editors, Photoshop
4. Forward and backward feature in web browsers

### **A.5 Procedure/Algorithm:**

#### **A.5.1:**

#### **TASK 1:**

Write a program to insert and delete from a specific position from an array

#### **TASK 2:**

Write a program to implement Stack (All the operations)

## **PART B**

**(PART B : TO BE COMPLETED BY STUDENTS)**

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

Roll No.: N036	Name:Nischaya Sharma
Class : MBA(Tech) Comp	Batch : B2
Date of Experiment: 13/07/19	Date of Submission:16/07/19
Grade :	Time of Submission:1:30 AM
Date of Grading:	

## B.1 Software Code written by student: (Task 1)

### Task1:

```
#include<stdio.h>
int main(){
    int num=10,arr[11],i,choice=5,position;
    for(i=0;i<11;i++){
        arr[i]=0;
    }
    for(i=0;i<10;i++){
        arr[i]=i;
    }
    printf("The array is:\n");
    for(i=0;i<10;i++){
        printf("%d\n",arr[i]);
    }
    while(choice!=0){
        printf("\nEnter: \n1. Enter an element\n2. Delete an element\n0. Exit\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:
                printf("Enter the position where you want to insert the element: ");
                scanf("%d",&position);
                for(i=9;i>=position-1;i--){
                    arr[i+1]=arr[i];
                }
                printf("Enter the element: ");
                scanf("%d",&arr[position-1]);
                num+=1;
                printf("The array is:\n");
                for(i=0;i<=num;i++){
                    printf("%d\n",arr[i]);
                }
                break;
            case 2:
                printf("Enter the position from where you want to delete the element: ");
                scanf("%d",&position);
                for(i=position-1;i<num;i++){
                    arr[i]=arr[i+1];
                }
                num-=1;
        }
    }
}
```

```

        printf("The array is:\n");
        for(i=0;i<=num;i++){
            printf("%d\n",arr[i]);
        }
        break;
    }
}
return 0;
}

```

## Task2:

```

class Stack {

    int max=1000;
    int[] arr = new int[max];
    int top = -1;
    Stack(int maximumElements){
        max = maximumElements;
        for(int i=0 ; i<maximumElements ; i++){
            arr[i]=-1;
        }
    }
    void peak(){
        System.out.println((top>=0)? ("The Topmost Element is "+arr[top]):"Underflow(The Stack is empty)");
    }
    void push(int num){
        if(top<max-1){
            arr[top+1]=num;
            top += 1;
        }
        else
            System.out.println("Overflow(The stack is full)");
    }
    void pop(){
        if(top>=0){
            arr[top]=-1;
            top -= 1;
        }
        else
            System.out.println("Underflow(The Stack is empty)");
    }
    void display(){
        for(int i=0 ; i<=top ; i++){
            System.out.print(arr[i]+" ", );
        }
    }
}

```

```

    }
    System.out.println();
}

//New Class Starts in a new file
import java.util.Scanner;
class Demo{
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter the length of the Stack you want to make: ");
        int max = in.nextInt();
        Stack stack = new Stack(max);
        int choice = 5;
        while(choice!=0){
            System.out.print("Enter \n1. Peak\n2. Display\n3. Push\n4. Pop\n0.
Exit\n");
            choice = in.nextInt();
            switch(choice){
                case 1:
                    stack.peak();
                    break;
                case 2:
                    stack.display();
                    break;
                case 3:
                    System.out.print("Enter the number to enter: ");
                    int num = in.nextInt();
                    stack.push(num);
                    stack.display();
                    break;
                case 4:
                    stack.pop();
                    stack.display();
                    break;
                case 0:
                    break;
                default:
                    System.out.println("Wrong Choice !!!!!");
                    break;
            }
        }
    }
}

```

## B.2 Input and Output: (Task 1)

### Task1:

```
enter limit5
enter values1
2
6
enter value to be inserted4
enter position4
1 2 3 4 5 6
enter value to be deleted5
1 2 3 4 6
Process returned 0 (0x0)   execution time : 19.765 s
Press any key to continue.
```

## Task2:

```
Enter the length of the Stack you want to make: 5
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
1
Underflow(The Stack is empty)
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
4
Underflow(The Stack is empty)

Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
3
Enter the number to enter: 1
1,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
3
Enter the number to enter: 2
1, 2,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
3
Enter the number to enter: 4
1, 2, 4,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
3
Enter the number to enter: 5
1, 2, 4, 5,
Enter
1. Peak
2. Display
3. Push
4. Pop
```

```
Enter the number to enter: 5
1, 2, 4, 5,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
3
Enter the number to enter: 6
1, 2, 4, 5, 6,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
3
Enter the number to enter: 7
Overflow(The stack is full)
1, 2, 4, 5, 6,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
1
The Topmost Element is 6
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
2
1, 2, 4, 5, 6,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
4
1, 2, 4, 5,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
4
1, 2, 4,
Enter
1. Peak
2. Display
3. Push
4. Pop
0. Exit
0
```

## B.3 Observations and learning [w.r.t. all tasks]:

*(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)*

## B.4 Conclusion:

***(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

## **B.5 Question of Curiosity**

***(To be answered by student based on the practical performed and learning/observations)***

### **Checking for palindrome string**

Suppose characters are arriving on a stream reader. Suggest an algorithm to see if the string forms a palindrome. Capitalization, spacing, and punctuation are ignored.

- 1) INSERT( STR, STR1)
- 2) [INITIALISE COUNTER] SET K = N, SET I = 0
- 3) [REPEAT 4,5,6] WHILE I < N, K >= 0
- 4) STR1[I] = STR[K]
- 5) [DECREASE COUNTER] SET K = K-1
- 6) [INCREASE COUNTER] SET I = I+1
- 7) [INITIALISE COUNTER] SET J = 0
- 8) [CHECK] IF STR[J] == STR1[J]
- 9) PALINDROME STRING
- 10) EXIT

\*\*\*\*\*