

# **Mechi Multiple Campus**

(Tribhuvan University)

Bhadrapur, Jhapa



## **Lab Report of**

**Data Structures and Algorithm (CACS-201)**

## **Implementation of STACK**

*Faculty of Humanities & Social Sciences*

Tribhuvan University

Kritipur, Nepal

## **Submitted By**

**Name:** Santosh Bhandari

**Roll No:** 58

## **Submitted To**

Mechi Multiple Campus

Department of Bachelor in Computer Application

Bhadrapur, Jhapa, Nepal

## Introduction to Stack

### Definition

A Stack is a linear collection of a data in which data are inserted and deleted from only one end called as top of Stack. Stack follows LIFO (Last In First Out) order i.e. the data inserted as the last will be the first one to be removed.

### Applications

- i) Revision
- ii) Arithmetic Calculations
- iii) Solving Graphs problems
- iv) Solving problem using backtracking techniques

### Terminologies

#### i) PUSH

The process of inserting data in the Stack.

#### ii) POP

The process of deleting data from the Stack.

#### iii) Peek

The process of identifying the data at the top of the Stack.

#### iv) Full Stack

If all the location of the Stack contains data.

#### v) Empty Stack

If the Stack doesn't contain any data.

#### vi) OVERFLOW

An attempt to push data in the full stack.

#### vii) Underflow

An attempt to pop data from an empty stack.

### Operations

#### i) PUSH Operation

Adding an element into the top of the Stack is referred to as push operation.

#### ii) POP Operation

Deletion of an element from the top of the Stack is called pop operation.

### iii) Peek or Display Operation

Display operation involves returning the element which is present at the top of the stack without deleting it.

### Algorithm to PUSH and POP data from the stack

#### PUSH operation Algorithm

- ① [check if the stack is full or Not]  
if  $TOS = MAX - 1$   
then print Overflow and exit
- ② [Increase the value of TOS by 1]  
 $TOS = TOS + 1$
- ③ [Insert the New data in TOS of the stack]  
 $STACK[TOS] = newdata$
- ④ EXIT

#### POP operation Algorithm

- ① [check if the stack is empty or Not]  
if  $TOS = -1$   
then print Underflow and exit.
- ② [Delete the data at the TOS]  
Delete  $STACK[TOS]$
- ③ [Decrease the value of TOS by 1]  
 $TOS = TOS - 1$
- ④ EXIT

## Program Code

```
#include<stdio.h>
void push();
void pop();
void display();
int stack[3],max=3,tos=-1;
void main(){
    top:
    printf("\n\n***Option***\n1. PUSH Data onto Stack\n2. POP Data from
Stack\n3. Display the Data of Stack\n\nEnter Your Option(1,2,3): ");
    int n;
    scanf("%d",&n);
    switch(n){
        case 1:
            push();
            goto top;
        case 2:
            pop();
            goto top;
        case 3:
            display();
            goto top;
        default:
            break;
    }
}
void push(){
    if(tos==(max-1))
        printf("OVERFLOW");
    else{
        tos++;
        printf("Enter the Number: ");
        scanf("%d",&stack[tos]);
        printf("%d is PUSH to Stack.",stack[tos]);
    }
}
void pop(){
    if(tos==-1)
        printf("UNDERFLOW");
    else{
        printf("%d is POP from Stack.",stack[tos]);
        tos--;
    }
}
void display(){
    if(tos==-1)
```



```

        printf("Stack is Empty.");
    else{
        int i;
        printf("Data on Stack: ");
        for(i=0;i<=tos;i++)
            printf("\t%d",stack[i]);
    }
}

```

## Output of the Program

```

***Option***
1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 1
Enter the Number: 10
10 is PUSH to Stack.

***Option***
1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 1
Enter the Number: 20
20 is PUSH to Stack.

***Option***
1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 1
Enter the Number: 30
30 is PUSH to Stack.

***Option***
1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 3
Data on Stack:  10      20      30

***Option***
1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 1
OVERFLOW

```

\*\*\*Option\*\*\*

1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 2  
30 is POP from Stack.

\*\*\*Option\*\*\*

1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 2  
20 is POP from Stack.

\*\*\*Option\*\*\*

1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 2  
10 is POP from Stack.

\*\*\*Option\*\*\*

1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of Stack

Enter Your Option(1,2,3): 3  
Stack is Empty.

\*\*\*Option\*\*\*

1. PUSH Data onto Stack
2. POP Data from Stack
3. Display the Data of stack

Enter Your Option(1,2,3): 2  
UNDERFLOW

### Conclusion

Hence, Stack is a collection of data in which data are inserted and deleted from only one end called Top of Stack. It follows Last In First Out order. The process of inserting data is push and deleting data is pop.

---