K. J. Somaiya College of Engineering, Mumbai

(A Constituent College of Somaiya Vidyavihar University)

Department of Computer Engineering

Batch: A2 Roll No.: 16010122041

Experiment / assignment / tutorial No.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Implementation of Stack applications.

Objective: To implement applications of stack

Expected Outcome of Experiment:

CO	Outcome
1	Explain the different data structures used in problem solving

Books/ Journals/ Websites referred:

1. *Fundamentals Of Data Structures In C* – Ellis Horowitz, Satraj Sahni, Susan Anderson-Fred
2. *An Introduction to data structures with applications* – Jean Paul Tremblay, Paul G. Sorenson
3. *Data Structures A Pseudo Approach with C* – Richard F. Gilberg & Behrouz A. Forouzan
4. https://www.cprogramming.com/tutorial/computersciencetheory/stack.html
5. https://www.geeksforgeeks.org/stack-data-structure-introduction-program/
6. https://www.thecrazyprogrammer.com/2013/12/c-program-for-array-representation-of-stack-push-pop-display.html

Assigned Stack application:
Reversal of String

Algorithm:

Step 1: Start

Step 2: Include Libraries

Include necessary libraries (stdio.h, stdlib.h, string.h).

Step 3: Define Maximum Stack Size

Set MAX_SIZE to 100, indicating the maximum stack size.

Step 4: Define Stack Functions

Create push, pop, and reverse functions.

Step 5: Initialize Stack Variables

Initialize top to -1 for an empty stack.Create str and strrev character arrays.

Step 6: Implement push Function

If the stack is full, print "Stack Overflow." Otherwise, increment top and add the character.

Step 7: Implement pop Function

If the stack is empty, print "Stack Underflow" and return '-'. Otherwise, retrieve the character at the top and decrement top.

Step 8: Implement reverse Function. Push characters from str onto the stack. Pop characters from the stack into strrev to reverse the string.

Step 9: Main Program

Prompt the user for a string. Reverse the input string using the reverse function. Display the reversed string.

Step 10: End

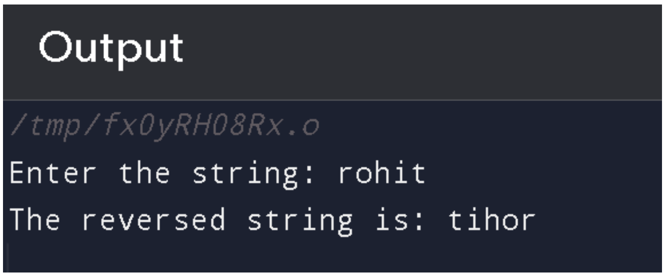
Example:
User input: rohit

Final output: tihor

Sourcecode:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX_SIZE 100

void push(char);
char pop();
void reverse();
static int top = -1;
char str[MAX_SIZE];
char strrev[MAX_SIZE];

void push(char data)
{
    if(top == MAX_SIZE-1)
    {
        printf("Stack Overflow\n");
        return;
    }
    else
    {
        top++;
        str[top] = data;
    }
}

char pop()
{
    if(top == -1)
    {
        printf("Stack Underflow\n");
        return '-';
    }
    else
    {
        char data = str[top];
        top--;
        return data;
    }
}

void reverse()
{
    for(int i=0; i<strlen(str); i++)
    {
        push(str[i]);
    }
    for(int i=0; i<strlen(str); i++)
    {
        strrev[i] = pop();
    }
}

int main()
{
    printf("Enter the string: ");
    scanf("%s", str);
    reverse();
    printf("The reversed string is: %s\n", strrev);
    return 0;
}
```

Output Screenshots:



```
/tmp/fxOyRHO8Rx.o
Enter the string: rohit
The reversed string is: tihor
```

Conclusion: We have learnt about the application of stacks in reversing a string. Stacks are a valuable tool that provides a strong foundation for understanding data structures and algorithms.