



COMPUTER ENGINEERING

DS ODD SEM 2021-22/EXPERIMENT 4

NAME:- GAURAV AMARNANI (D7A, 67)

Assignment - #4

AIM:- Applications of Stack ADT Reversing a string

Theory:-

A stack is a linear data structure in which insertion and deletion of elements are done at only one end which is known as the top of the stack. Stack is called a last in first out (LIFO) structure because the last element which is added to the stack is the first element which is deleted from stack. Stack are used where the LIFO principle is required like reversing strings, checking whether the arithmetic expression is properly parenthesized, converting infix notation to postfix and prefix notation, evaluating postfix expression, implementing recursions, various applications and function calls etc. Various applications of stacks are listed below and this section discusses some of these applications.

- 1) Reversing a string
- 2) Polish notations
- 3) Conversion of Infix to postfix expression
- 4) Evaluation of Postfix expression
- 5) Conversion of Infix to Prefix expression
- 6) Evaluation of Prefix expression
- 7) Recursion
- 8) Backtracking

Stacks are widely used in operating systems, by compiler and by applications. Some of the applications are.

- 1/ Subroutine calls.
- 2/ Interrupt handling
- 3/ Matching parentheses in an expression.

A simple application of stacks is reversing strings. This can be achieved very easily by reading the input string character by character and push that onto stack, till end of string is reached. Once all the characters of the string are pushed onto the stack, they are popped one by one. Since the character last pushed in comes out first, subsequent pop operations result in reversal of the string.

Algorithm:

Step 1: Set $i = 0$

Step 2: While ($i < \text{length_of_str}$)
 Push $\text{str}[i]$ onto the stack
 Set $i + 1$
End while

Step 3: Set $i = \text{TOP}$

Step 4: While ($i >= 0$)
 Pop the TOP element of the stack
 and store it in $\text{str}[i]$

Set $i = i - 1$

End While.

Step 5: Print "The reversed string is ", str

Step 6: Exit

Conclusion: From this experiment we implemented the program of reversing a string using stack and also learnt about the various applications and implementation of stack.

PROGRAM:

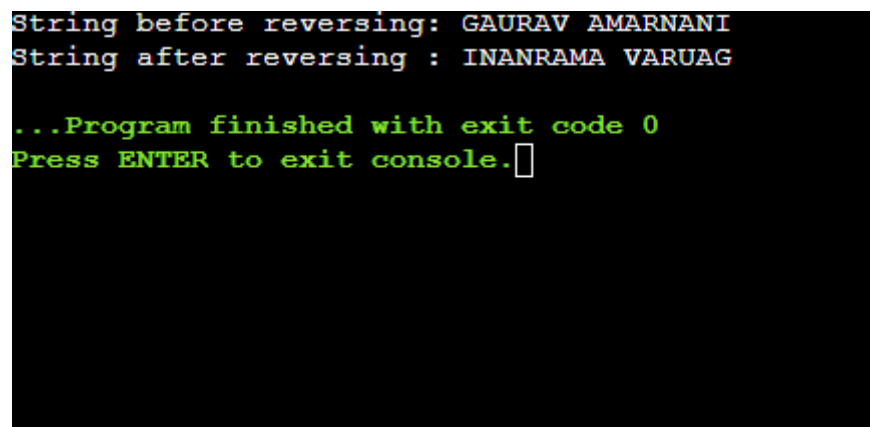
```
#include <stdio.h>
#include <string.h>
#define max 100
int top,stack[max];

void push(char x){
if(top == max-1){
printf("stack overflow");
}
else {
}
stack[++top]=x;
}
void pop(){
printf("%c",stack[top--]);
}

void main() {
char str[]="KESAR";
int len = strlen(str);
int i;

clrscr();
printf("String before reversing : %s\n", &str);
for(i=0;i<len;i++)
push(str[i]);
printf("String after reversing : ");
for(i=0;i<len;i++)
pop();
getch();
}
```

OUTPUT:-

A screenshot of a terminal window with a black background and green text. The output shows the string 'GAURAV AMARNANI' being reversed to 'INANRAMA VARUAG'. It also shows the program finishing with exit code 0 and a prompt to press ENTER to exit the console.

```
String before reversing: GAURAV AMARNANI
String after reversing : INANRAMA VARUAG

...Program finished with exit code 0
Press ENTER to exit console. □
```