

# Rajalakshmi Engineering College

Name: Sanjai E  
Email: 241801242@rajalakshmi.edu.in  
Roll no: 241801242  
Phone: 9363574090  
Branch: REC  
Department: I AI & DS FD  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 3\_COD\_Question 3

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Sharon is developing a programming challenge for a coding competition. The challenge revolves around implementing a character-based stack data structure using an array.

Sharon's project involves a stack that can perform the following operations:

Push a Character: Users can push a character onto the stack. Pop a Character: Users can pop a character from the stack, removing and displaying the top character. Display Stack: Users can view the current elements in the stack. Exit: Users can exit the stack operations application.

Write a program to help Sharon to implement a program that performs the given operations.

***Input Format***

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the character onto the stack. If the choice is 1, the following input is a space-separated character, representing the character to be pushed onto the stack.

Choice 2: Pop the character from the stack.

Choice 3: Display the characters in the stack.

Choice 4: Exit the program.

### ***Output Format***

The output displays messages according to the choice and the status of the stack:

1. If the choice is 1, push the given character to the stack and display the pushed character having the prefix "Pushed: ".
2. If the choice is 2, undo the character from the stack and display the character that is popped having the prefix "Popped: ".
3. If the choice is 2, and if the stack is empty without any characters, print "Stack is empty. Nothing to pop."
4. If the choice is 3, print the elements in the stack having the prefix "Stack elements: ".
5. If the choice is 3, and there are no characters in the stack, print "Stack is empty."
6. If the choice is 4, exit the program.
7. If any other choice is entered, print "Invalid choice"

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 2

4

Output: Stack is empty. Nothing to pop.

### ***Answer***

```
#include <stdio.h>
```

```
#include <stdbool.h>
```

```
#define MAX_SIZE 100
```

```
char items[MAX_SIZE];
```

```
int top = -1;
```

```
void initialize() {
```

```
    top = -1;
```

```
}
```

```
bool isFull() {
```

```
    return top == MAX_SIZE - 1;
```

```
}
```

```
bool isEmpty() {
```

```
    return top == -1;
```

```
}
```

```
void push(char ch) {
```

```
    if (!isFull()) {
```

```
        items[++top] = ch;
```

```
        printf("Pushed: %c\n", ch);
```

```
    }
```

```
}
```

```
void pop() {
```

```
    if (!isEmpty()) {
```

```
        printf("Popped: %c\n", items[top--]);
```

```
    } else {
```

```
        printf("Stack is empty. Nothing to pop.\n");
```

```
    }
```

```
}
```

```
void display() {
```

```
    if (!isEmpty()) {
```

```
        printf("Stack elements: ");
```

```
        for (int i = top; i >= 0; i--) {
```

```
            printf("%c ", items[i]);
```

```
        }
```

```
        printf("\n");
```

```
    } else {
```

```
        printf("Stack is empty.\n");
    }
}

int main() {
    initialize();
    int choice;
    char value;

    while (true) {
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                scanf(" %c", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                return 0;
            default:
                printf("Invalid choice\n");
        }
    }
    return 0;
}
```

**Status :** Correct

**Marks :** 10/10