

# Rajalakshmi Engineering College

Name: tharunika R

Email: 241801296@rajalakshmi.edu.in

Roll no: 241801296

Phone: 6369646218

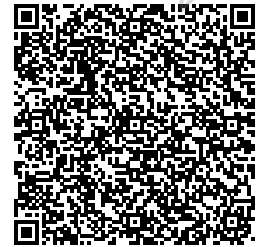
Branch: REC

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 5

Attempt : 1

Total Mark : 10

Marks Obtained : 10

### Section 1 : Coding

#### 1. Problem Statement

Milton is a diligent clerk at a school who has been assigned the task of managing class schedules. The school has various sections, and Milton needs to keep track of the class schedules for each section using a stack-based system.

He uses a program that allows him to push, pop, and display class schedules for each section. Milton's program uses a stack data structure, and each class schedule is represented as a character. Help him write a program using a linked list.

#### ***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 class schedule to be pushed onto the stack.

Choice 2: Pop class schedule from the stack

Choice 3: Display the class schedules in the stack.

Choice 4: Exit the program.

### ***Output Format***

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

- If the choice is 1, push the given class schedule to the stack and display the following: "Adding Section: [class schedule]"
- If the choice is 2, pop the class schedule from the stack and display the following: "Removing Section: [class schedule]"
- If the choice is 2, and if the stack is empty without any class schedules, print "Stack is empty. Cannot pop."
- If the choice is 3, print the class schedules in the stack in the following: "Enrolled Sections: " followed by the class schedules separated by space.
- If the choice is 3, and there are no class schedules in the stack, print "Stack is empty"
- If the choice is 4, exit the program and display the following: "Exiting the program"
- If any other choice is entered, print "Invalid choice"

Refer to the sample output for the exact format.

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

Input: 1 d

1 h

3

2

3

4

Output: Adding Section: d

Adding Section: h

Enrolled Sections: h d

Removing Section: h

Enrolled Sections: d

Exiting program

### **Answer**

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

```
#include <stdlib.h>
```

```
struct Node {  
    char data;  
    struct Node* next;  
};
```

```
struct Node* top = NULL;
```

```
// You are using Java  
import java.util.Scanner;
```

```
class Node {  
    char data;  
    Node next;  
  
    Node(char data) {  
        this.data = data;  
        this.next = null;  
    }  
}
```

```
class Stack {  
    private Node top;
```

```
    Stack() {  
        this.top = null;  
    }
```

```
    public void push(char data) {  
        Node newNode = new Node(data);
```

```
newNode.next = top;
top = newNode;
System.out.println("Adding Section: " + data);
}
```

```
public void pop() {
    if (top == null) {
        System.out.println("Stack is empty. Cannot pop.");
    } else {
        char removed = top.data;
        top = top.next;
        System.out.println("Removing Section: " + removed);
    }
}
```

```
public void display() {
    if (top == null) {
        System.out.println("Stack is empty");
    } else {
        System.out.print("Enrolled Sections: ");
        Node current = top;
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println();
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Stack stack = new Stack();
```

```
        while (true) {
            if (!scanner.hasNextInt()) break;
            int choice = scanner.nextInt();
```

```
            switch (choice) {
                case 1:
                    if (!scanner.hasNext()) break;
```

```

        char ch = scanner.next().charAt(0);
        if (Character.isLetter(ch)) {
            stack.push(ch);
        }
        break;

    case 2:
        stack.pop();
        break;

    case 3:
        stack.display();
        break;

    case 4:
        System.out.println("Exiting program");
        return;

    default:
        System.out.println("Invalid choice");
    }
}

```

```

        scanner.close();
    }
}

```

```

int main() {
    int choice;
    char value;
    do {
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                scanf(" %c", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                displayStack();
                break;

```

```
        case 4:
            printf("Exiting program\n");
            break;
        default:
            printf("Invalid choice\n");
    }
} while (choice != 4);

return 0;
}
```

**Status :** Correct

**Marks : 10/10**