Rajalakshmi Engineering College

Name: Jayasree D
Email: 241801100@rajalakshmi.edu.in
Roll no:
Phone: 9025821157
Branch: REC
Department: I AI & DS FB
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 Python
class Node:
   def __init__(self, data):
      self.data = data
      self.next = None

class Stack:
   def __init__(self):
      self.top = None

   def push(self, data):
      if not data.isalpha():
         print("Invalid input. Only alphabetic characters are allowed.")
         return

      new_node = Node(data)
      new_node.next = self.top
      self.top = new_node
      print(f"Adding Section: {data}")

   def pop(self):
      if self.top is None:
```

```
            print("Stack is empty. Cannot pop.")
            return

        removed_data = self.top.data
        self.top = self.top.next
        print(f"Removing Section: {removed_data}")

    def display(self):
        if self.top is None:
            print("Stack is empty")
            return

        current = self.top
        print("Enrolled Sections: ", end="")
        while current:
            print(current.data, end=" ")
            current = current.next
        print()

def main():
    stack = Stack()

    while True:
        choice = input().strip()

        if choice.startswith("1 "):
            _, data = choice.split()
            stack.push(data)
        elif choice == "2":
            stack.pop()
        elif choice == "3":
            stack.display()
        elif choice == "4":
            print("Exiting program")
            break
        else:
            print("Invalid choice")

if __name__ == "__main__":
    main()
```

```
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***