

Rajalakshmi Engineering College

Name: Parvendhan C
Email: 241801197@rajalakshmi.edu.in
Roll no: 241801197
Phone: 8270861183
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

You are required to implement a stack data structure using a singly linked list that follows the Last In, First Out (LIFO) principle.

The stack should support the following operations: push, pop, display, and peek.

Input Format

The input consists of four space-separated integers N, representing the elements to be pushed onto the stack.

Output Format

The first line of output displays all four elements in a single line separated by a space.

The second line of output is left blank to indicate the pop operation without displaying anything.

The third line of output displays the space separated stack elements in the same line after the pop operation.

The fourth line of output displays the top element of the stack using the peek operation.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 11 22 33 44

Output: 44 33 22 11

33 22 11

33

Answer

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

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

```
typedef struct Node {  
    int data;  
    struct Node* next;  
} Node;
```

```
Node* createNode(int data) {  
    Node* newNode = (Node*)malloc(sizeof(Node));  
    newNode->data = data;  
    newNode->next = NULL;  
    return newNode;  
}
```

```
void push(Node** top, int data) {  
    Node* newNode = createNode(data);  
    newNode->next = *top;  
    *top = newNode;  
}
```

```
void pop(Node** top) {  
    if (*top == NULL) return;  
    Node* temp = *top;  
    *top = (*top)->next;  
    free(temp);  
}
```

```
int peek(Node* top) {  
    if (top == NULL) return -1;  
    return top->data;  
}
```

```
void display(Node* top) {  
    Node* temp = top;  
    while (temp != NULL) {  
        printf("%d ", temp->data);  
        temp = temp->next;  
    }  
    printf("\n");  
}
```

```
int main() {  
    Node* stack = NULL;  
    int a, b, c, d;  
  
    scanf("%d %d %d %d", &a, &b, &c, &d);  
  
    push(&stack, a);  
    push(&stack, b);  
    push(&stack, c);  
    push(&stack, d);  
  
    display(stack);  
  
    pop(&stack);  
    printf("\n");  
  
    display(stack);  
    printf("%d\n", peek(stack));  
}
```

```
    return 0;  
}
```

Status : Correct

Marks : 10/10

2. Problem Statement

Rithi is building a simple text editor that allows users to type characters, undo their typing, and view the current text. She has implemented this text editor using an array-based stack data structure.

She has to develop a basic text editor with the following features:

Type a Character (Push): Users can type a character and add it to the text editor. Undo Typing (Pop): Users can undo their typing by removing the last character they entered from the editor. View Current Text (Display): Users can view the current text in the editor, which is the sequence of characters in the buffer. Exit: Users can exit the text editor application.

Write a program that simulates this text editor's undo feature using a character stack and implements the push, pop and display operations accordingly.

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: