**Academic Year: 2023-24 Name of Student:**

**Semester: III Student ID:**

**Class / Branch/ Div: SE- IT A/ B/ C Roll No.**

**Subject: Data Structure Lab Date of Submission:**

**Name of Instructor:**

**Experiment No.2**

**Aim: Implementation of Stack and Queue using Linked List.**

# Code:

**#include <stdio.h>**

**#include <stdlib.h>**

**void push();**

**void pop();**

**void display();**

**struct node**

**{**

**int val;**

**struct node \*next;**

**};**

**struct node \*head;**

**void main ()**

**{**

**int choice=0;**

**printf("\n\*\*\*\*\*\*\*\*\*Stack operations using linked list\*\*\*\*\*\*\*\*\*\n");**

**printf("\n----------------------------------------------\n");**

**while(choice != 4)**

**{**

**printf("\n\nChose one from the below options...\n");**

**printf("\n1.Push\n2.Pop\n3.Show\n4.Exit");**

**printf("\n Enter your choice \n");**

**scanf("%d",&choice);**

**switch(choice)**

**{**

**case 1:**

**{**

**push();**

**break;**

**}**

**case 2:**

**{**

**pop();**

**break;**

**}**

**case 3:**

**{**

**display();**

**break;**

**}**

**case 4:**

**{**

**printf("Exiting....");**

**break;**

**}**

**default:**

**{**

**printf("Please Enter valid choice ");**

**}**

**};**

**}**

**}**

**void push ()**

**{**

**int val;**

**struct node \*ptr = (struct node\*)malloc(sizeof(struct node));**

**if(ptr == NULL)**

**{**

**printf("not able to push the element");**

**}**

**else**

**{**

**printf("Enter the value");**

**scanf("%d",&val);**

**if(head==NULL)**

**{**

**ptr->val = val;**

**ptr -> next = NULL;**

**head=ptr;**

**}**

**else**

**{**

**ptr->val = val;**

**ptr->next = head;**

**head=ptr;**

**}**

**printf("Item pushed");**

**}**

**}**

**void pop()**

**{**

**int item;**

**struct node \*temp;**

**if (head == NULL)**

**{**

**printf("Underflow");**

**}**

**else**

**{**

**item = head->val;**

**temp = head;**

**head = head->next;**

**free(temp);**

**printf("Item popped");**

**}**

**}**

**void display()**

**{**

**int i;**

**struct node \*ptr;**

**ptr=head;**

**if(ptr == NULL)**

**{**

**printf("Stack is empty\n");**

**}**

**else**

**{**

**printf("Printing Stack elements \n");**

**while(ptr!=NULL)**

**{**

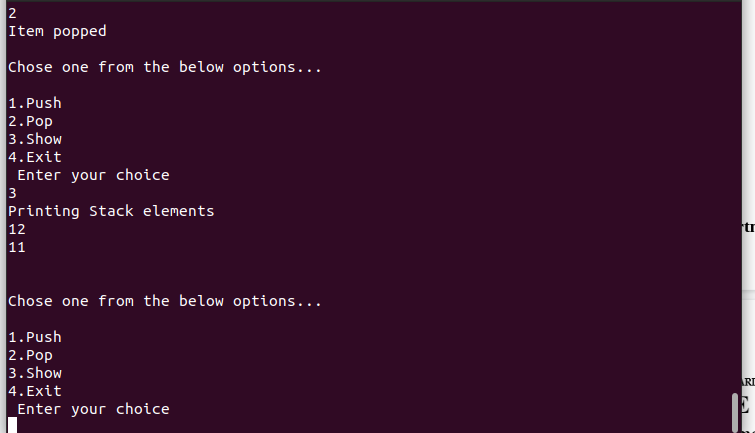
**printf("%d\n",ptr->val);**

**ptr = ptr->next;**

**}**

**}**

**}**

****

**Queue using Linked List**

**Code:**

**#include<stdio.h>**

**#include <stdlib.h>**

**struct Node**

**{**

**int data;**

**struct Node \*next;**

**}\*front = NULL,\*rear = NULL;**

**void insert(int);**

**void delete();**

**void display();**

**void main()**

**{**

**int choice, value;**

**printf("\n:: Queue Implementation using Linked List ::\n");**

**while(1){**

**printf("\n\*\*\*\*\*\* MENU \*\*\*\*\*\*\n");**

**printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n"); printf("Enter your choice: ");**

**scanf("%d",&choice); switch(choice){**

**case 1: printf("Enter the value to be insert: ");**

**scanf("%d", &value);**

**insert(value);**

**break;**

**case 2: delete();**

**break;**

**case 3: display();**

**break;**

**case 4:**

**exit(0);**

**default:**

**printf("\nWrong selection!!! Please try again!!!\n");**

**}**

**}**

**}**

**void insert(int value)**

**{**

**struct Node \*newNode;**

**newNode = (struct Node\*)malloc(sizeof(struct Node));**

**newNode->data = value;**

**newNode -> next = NULL;**

**if(front == NULL)**

**front = rear = newNode;**

**else{**

**rear -> next = newNode;**

**rear = newNode;**

**}**

**printf("\nInsertion is Success!!!\n");**

**}**

**void delete()**

**{**

**if(front == NULL)**

**printf("\nQueue is Empty!!!\n");**

**else{**

**struct Node \*temp = front;**

**front = front -> next;**

**printf("\nDeleted element: %d\n", temp->data);**

**free(temp);**

**}**

**}**

**void display()**

**{**

**if(front == NULL)**

**printf("\nQueue is Empty!!!\n");**

**else**

**{**

**struct Node \*temp = front;**

**while(temp->next != NULL)**

**{**

**printf("%d--->",temp->data);**

**temp = temp -> next;**

**}**

**printf("%d--->NULL\n",temp->data);**

**}**

**}**

****

# Conclusion: -Thus, we have implemented stack and queue data structure using linked list. In the linked list we dynamically allocate the memory to the node and keep the track of the next node.