

SANGHAMITRA R

LAB 5

6) WAP to Implement Single Link List to simulate Stack & Queue Operations

STACK:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head = NULL;
void push(int val)
{
    struct node *newNode = malloc(sizeof(struct node));
    newNode->data = val;
    newNode->next = head;
    head = newNode;
}
void pop()
{
    struct node *temp;
    if(head == NULL)
        printf("Stack is Empty\n");
    else
    {
        printf("Popped element = %d\n", head->data);
        temp = head;
        head = head->next;
        free(temp);
    }
}

void printList()
{
    struct node *temp = head;
    while(temp != NULL)
    {
```

```

printf("%d->", temp->data);
temp = temp->next;
}
printf("NULL\n");
}
int main()
{
int data, ch;
printf("Menu:\n 1. Push\n 2. Pop\n 3. Display\n 4. Exit");
printf("\nEnter choice: ");
scanf("%d", &ch);
while(ch!=4){
switch(ch){
case 1:
printf("Enter data to be pushed: ");
scanf("%d", &data);
push(data);
break;
case 2:
pop();
break;
case 3:
printList();
break;
case 4:
exit(0);
}
printf("\nEnter choice: ");
scanf("%d", &ch);
}
return 0;
}

```

Output:

Menu:

1. Push
2. Pop
3. Display
4. Exit

Enter choice: 1

Enter data to be pushed: 12

Enter choice: 1

Enter data to be pushed: 23

Enter choice: 1

Enter data to be pushed: 45

Enter choice: 2

Popped element = 45

Enter choice: 3

23->12->NULL

Enter choice: 4

PS C:\vs code files\C tutorials> █

QUEUE

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *front = NULL, *rear = NULL;
void enqueue(int val)
{
    struct node *newNode = malloc(sizeof(struct node));
    newNode->data = val;
    newNode->next = NULL;
    if(front == NULL && rear == NULL)
```

```

front = rear = newNode;
else
{
    rear->next = newNode;

    rear = newNode;
}
}

void dequeue()
{
    struct node *temp;
    if(front == NULL)
        printf("Queue is Empty. Unable to perform dequeue\n");
    else
    {
        temp = front;
        front = front->next;
        if(front == NULL)
            rear = NULL;
        free(temp);
    }
}

void printList()
{
    struct node *temp = front;
    while(temp)
    {
        printf("%d->", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main()
{
    int data, ch;
    printf("Menu:\n 1. Enqueue\n 2. Dequeue\n 3. Display\n 4. Exit");
    printf("\nEnter choice: ");
    scanf("%d", &ch);
}

```

```
while(ch!=4) {
    switch(ch) {
        case 1:
            printf("Enter data to be pushed: ");
            scanf("%d", &data);
            enqueue(data);
            break;
        case 2:
            dequeue();
            break;
        case 3:
            printList();
            break;
        case 4:
            exit(0);
    }
    printf("\nEnter choice: ");
    scanf("%d", &ch);
}
return 0;
}
```

Output:

Menu:

1. Enqueue
2. Dequeue
3. Display
4. Exit

Enter choice: 1

Enter data to be pushed: 12

Enter choice: 1

Enter data to be pushed: 23

Enter choice: 1

Enter data to be pushed: 45

Enter choice: 1

Enter data to be pushed: 67

Enter choice: 2

Enter choice: 2

Enter choice: 3

45->67->NULL

Enter choice: 4

PS C:\vs code files\C tutorials> █