

Name: Vinayak Kumar Singh

Register No: 23MCA1030

1.Implementation of stack using Linked list.

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

struct Node
{
    int data;
    struct Node *next;
};
struct Node *top = NULL;

void push(int);
void pop();
void display();

int main()
{
    int choice, value;
    printf("\nImplementation of Stack using LinkedList");
    while (1)
    {
        printf("\nChoose from available options");
        printf("\n1.Push \n2.Pop \n3.Display \n4.Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
                printf("Enter the input: ");
                scanf("%d", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                exit(0);
            default:
```

```

        printf("\nPlease choose from available options");
    }
}
return 0;
}

void push(int value)
{
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->data = value;

    if (top == NULL)
        newNode->next = NULL;
    else
        newNode->next = top;

    top = newNode;
    printf("Input Inserted");
}

void pop()
{
    if (top == NULL)
        printf("Stack has no value\n");
    else
    {
        struct Node *temp = top;
        printf("Last element Deleted %d", temp->data);
        top = temp->next;
        free(temp);
    }
}

void display()
{
    if (top == NULL)
        printf("Stack has no value\n");
    else
    {
        struct Node *temp = top;
        while (temp->next != NULL)
        {
            printf("%d--->", temp->data);
            temp = temp->next;
        }
        printf("%d--->NULL", temp->data);
    }
}

```

Output:

PROBLEMS OUTPUT TERMINAL

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.
```

```
D:\MCA\2. Data Structures and Algorithms + LAB>cd "d:\MCA\2. Data Structures and Algorithms + LAB\DSA\" && gcc stacklinkedlist.c -o stacklinkedlist && "d:\MCA\2. Data Structures and Algorithms + LAB\DSA\"stacklinkedlist
```

```
Implementation of Stack using LinkedList
Choose from available options
```

```
1.Push
```

```
2.Pop
```

```
3.Display
```

```
4.Exit
```

```
Enter your choice: 1
```

```
Enter the input: 10
```

```
Input Inserted
```

```
Choose from available options
```

```
1.Push
```

```
2.Pop
```

```
3.Display
```

```
4.Exit
```

```
Enter your choice: 1
```

```
Enter the input: 20
```

```
Input Inserted
```

```
Choose from available options
```

```
1.Push
```

```
2.Pop
```

```
3.Display
```

```
4.Exit
```

```
Enter your choice: 1
```

```
Enter the input: 30
```

```
Input Inserted
```

```
Choose from available options
```

```
1.Push
```

```
2.Pop
```

```
3.Display
```

```
4.Exit
```

```
Enter your choice: 3
```

```
30--->20--->10--->NULL
```

```
Choose from available options
```

```
1.Push
```

```
2.Pop
```

```
3.Display
```

```
4.Exit
```

```
Enter your choice: 2
```

```
Last element Deleted 30
```

Choose from available options

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 3

20--->10--->NULL

Choose from available options

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 4

d:\MCA\2. Data Structures and Algorithms + LAB\DSA>

2.Implementation of queue using Linked list.

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

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

struct Queue
{
    struct Node *front;
    struct Node *rear;
};

void insert(struct Queue *queue, int input)
{
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->data = input;
    newNode->next = NULL;

    if (queue->front == NULL)
    {
        queue->front = queue->rear = newNode;
    }
    else
    {
        queue->rear->next = newNode;
        queue->rear = newNode;
    }
}
```

```

    }
    printf("Input Inserted");
}

void delete(struct Queue *queue)
{
    if (queue->front == NULL)
    {
        printf("Queue has no values\n");
    }
    else
    {
        struct Node *temp = queue->front;
        queue->front = queue->front->next;
        printf("Front element Deleted: %d", temp->data);
        free(temp);
    }
}

void display(struct Queue *queue)
{
    if (queue->front == NULL)
    {
        printf("Queue has no values");
    }
    else
    {
        struct Node *temp = queue->front;
        while (temp != NULL)
        {
            printf("%d--->", temp->data);
            temp = temp->next;
        }
        printf("NULL");
    }
}

int main()
{
    struct Queue queue = {NULL, NULL};
    int choice, input;

    printf("\nImplementation of queue using Linked List");
    while (1)
    {
        printf("\nChoose from available options");
        printf("\n1.Insert \n2.Delete \n3.Display \n4.Exit");
        printf("\nEnter your choice: ");
    }
}

```

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

switch (choice)
{
case 1:
    printf("Enter the input: ");
    scanf("%d", &input);
    insert(&queue, input);
    break;
case 2:
    delete (&queue);
    break;
case 3:
    display(&queue);
    break;
case 4:
    exit(0);
default:
    printf("\nPlease choose from available options");
}
}
return 0;
}
```

Output:

PROBLEMS OUTPUT TERMINAL

Microsoft Windows [Version 10.0.22621.2134]

(c) Microsoft Corporation. All rights reserved.

D:\MCA\2. Data Structures and Algorithms + LAB>cd "d:\MCA\2. Data Structures and Algorithms + LAB\DSA\" && gcc queuesusinglinkedlist.c -o queuesusinglinkedlist && "d:\MCA\2. Data Structures and Algorithms + LAB\DSA\"queuesusinglinkedlist

Implementation of queue using Linked List

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 3

Queue has no values

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 1

Enter the input: 10

Input Inserted

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 1

Enter the input: 20

Input Inserted

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 1

Enter the input: 30

Input Inserted

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 2

Front element Deleted: 10

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 3

20--->30--->NULL

Choose from available options

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice: 4

d:\MCA\2. Data Structures and Algorithms + LAB\DSA>

END