

```

// Write a program in C to implement Stack and its various operations
using linked list.
// The various operations with the stack are:
// a)Create a stack
// b)Check whether it is empty or not?
// c)Check whether it is full or not?
// d)Insert operation (PUSH operation) with the stack.
// e)Delete operation (POP operation) with the stack.
#include <stdio.h>
#include <stdlib.h>

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

void create();
// void isFull();
void isEmpty();
void push(int value);
int pop();
void display();

int main()
{
    int ch, value;
    printf("\nImplementaion of Stack using Linked List\n");
    while (1)
    {
        printf("\t1) Create a stack\n");
        printf("\t2) Check wheather stack is full\n");
        printf("\t3) Check wheather stack is empty\n");
        printf("\t4) Insert operation in stack\n");
        printf("\t5) Delete operation in stack\n");
        printf("\t6) Display the stack\n");
        printf("\t7) Exit\n");

        printf("\nEnter your choice : ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:
                create();
                break;

```

```

        case 2:
            // isFull();
            break;
        case 3:
            isEmpty();
            break;
        case 4:
            printf("\nEnter the value to insert: ");
            scanf("%d", &value);
            push(value);
            break;
        case 5:
            printf("Popped element is :%d\n", pop());
            break;
        case 6:
            display();
            break;
        case 7:
            exit(0);
            break;
        default:
            printf("\nWrong Choice\n");
    }
}

void create()
{
    int val;
    char ch;
    // printf("Enter the size of stack you want\n");
    // scanf("%d",&size);
    printf("Enter the value to be inserted\n");
    scanf("%d", &val);
    push(val);
    printf("Enter anykey to continue or q to exit\n");
    scanf(" %c", &ch);
    while (ch != 'q')
    {
        printf("Enter the value to be inserted\n");
        scanf("%d", &val);
        push(val);
        printf("Enter anykey to continue or q to exit\n");
        scanf(" %c", &ch);
    }
}

```

```

// void isFull()
// {
//     if (!node)
//     {
//         printf("Stack is full\n");
//         exit(-1);
//     }
// }
void isEmpty()
{
    if (top == NULL)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("Stack is not empty\n");
    }
}
void push(int value)
{
    struct node *newNode;
    newNode = (struct node *)malloc(sizeof(struct node));
    newNode->data = value;
    if (top == NULL)
    {
        newNode->next = NULL;
    }
    else
    {
        newNode->next = top; // Make the node as top
    }
    top = newNode; // top always points to the newly created node
    printf("node is Inserted\n\n");
}

int pop()
{
    if (top == NULL)
    {
        printf("\nStack Underflow\n");
    }
    else
    {
        struct node *temp = top;
        int temp_data = top->data;
    }
}

```

```

        top = top->next;
        free(temp);
        return temp_data;
    }
}

void display()
{
    if (top == NULL)
    {
        printf("\nStack Underflow\n");
    }
    else
    {
        printf("The stack is \n");
        struct node *temp = top;
        while (temp->next != NULL)
        {
            printf("%d\t", temp->data);
            temp = temp->next;
        }
        printf("%d\n\n", temp->data);
    }
}

```

// Write a program in C to implement Queue and its various operations using linked list.

// The operations with the queue are:

// a)Create a queue

// b)Check whether it is empty or not?

// c)Check whether it is full or not?

// d)Insert operation with the queue.

// e)Delete operation with the queue.

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

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

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node *next;
```

```

};

struct node *front = NULL;
struct node *rear = NULL;

void create();
// void isFull();
void isEmpty();
void insert(int value);
int delete();
void display();

int main()
{
    int ch, value;
    printf("\nImplementaion of Stack using Linked
List\n");
    while (1)
    {
        printf("\t1) Create a queue\n");
        printf("\t2) Check wheather queue is
full\n");
        printf("\t3) Check wheather queue is
empty\n");
        printf("\t4) Insert operation in queue\n");
        printf("\t5) Delete operation in queue\n");
        printf("\t6) Display the queue\n");
        printf("\t7) Exit\n");

        printf("\nEnter your choice : ");
        scanf("%d", &ch);
        switch (ch)
        {

```

```

        case 1:
            create();
            break;
        case 2:
            // isFull();
            break;
        case 3:
            isEmpty();
            break;
        case 4:
            printf("\nEnter the value to insert:
");
            scanf("%d", &value);
            insert(value);
            break;
        case 5:
            printf("Deleted element is :%d\n",
delete());
            break;
        case 6:
            display();
            break;
        case 7:
            exit(0);
            break;
        default:
            printf("\nWrong Choice\n");
    }
}

void create()
{
    int val;

```

```

    char ch;
    // printf("Enter the size of stack you
want\n");
    // scanf("%d",&size);
    printf("Enter the value to be inserted\n");
    scanf("%d", &val);
    insert(val);
    printf("Enter anykey to continue or q to
exit\n");
    scanf(" %c", &ch);
    while (ch != 'q')
    {
        printf("Enter the value to be inserted\n");
        scanf("%d", &val);
        insert(val);
        printf("Enter anykey to continue or q to
exit\n");
        scanf(" %c", &ch);
    }
}
void isEmpty()
{
    if ((front == NULL) && (rear==NULL))
    {
        printf("Queue is empty\n");
    }
    else
    {
        printf("Queue is not empty\n");
    }
}
void insert(int value)
{

```

```

    struct node *ptr;
    ptr = (struct node *)malloc(sizeof(struct
node));
    ptr->data = value;
    ptr->next = NULL;
    if ((front == NULL) && (rear == NULL))
    {
        front = rear = ptr;
    }
    else
    {
        rear->next = ptr;
        rear = ptr;
    }
    printf("Node is Inserted\n\n");
}

int delete()
{
    if (front == NULL)
    {
        printf("\nUnderflow\n");
        return -1;
    }
    else
    {
        struct node *temp = front;
        int temp_data = front->data;
        front = front->next;
        free(temp);
        return temp_data;
    }
}

```



```
void display()
{
    struct node *temp;
    if ((front == NULL) && (rear == NULL))
    {
        printf("\nQueue is Empty\n");
    }
    else
    {
        printf("The queue is \n");
        temp = front;
        while (temp) {
            printf("%d\t", temp->data);
            temp = temp->next;
        }
        printf("NULL\n\n");
    }
}
```