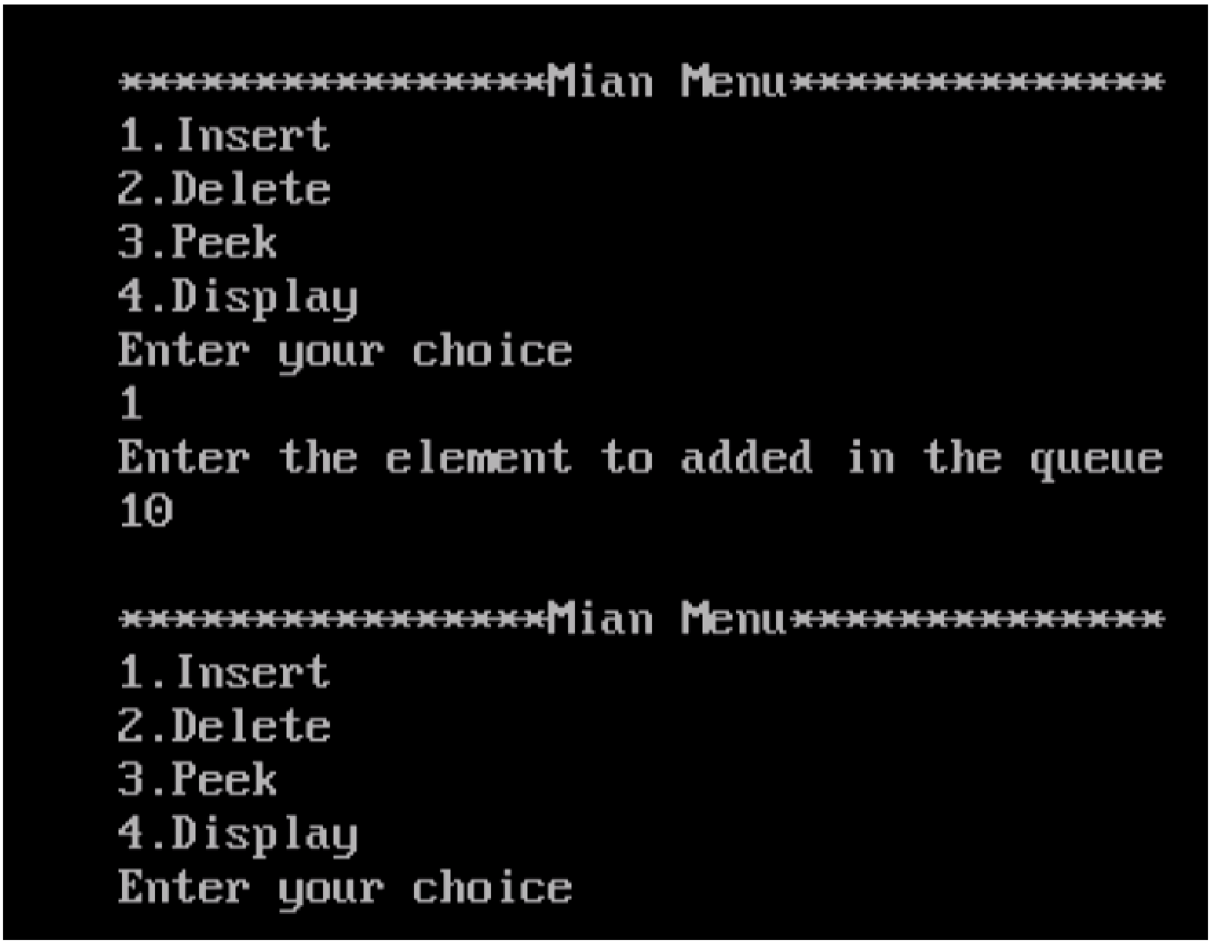
SOURCE CODE: LINKED QUEUE:

```
#include<stdio.h>
#include<conio.h>
struct node
{
        int data;
        struct node *next;
};
struct queue
{
        struct node *front;
        struct node *rear;
};
struct queue *q;
void create_queue(struct queue *);
struct queue *insert(struct queue *,int);
struct queue *delete_element(struct queue *);
struct queue *display(struct queue *);
int peek(struct queue *);
void main()
{
        int val,ch;
        clrscr();
        do
        {
                printf("\n****************Mian Menu**************\n");
                printf("1.Insert\n2.Delete\n3.Peek\n4.Display\n");
                printf("Enter your choice\n");
                scanf("%d",&ch);
                switch(ch)
                {
```

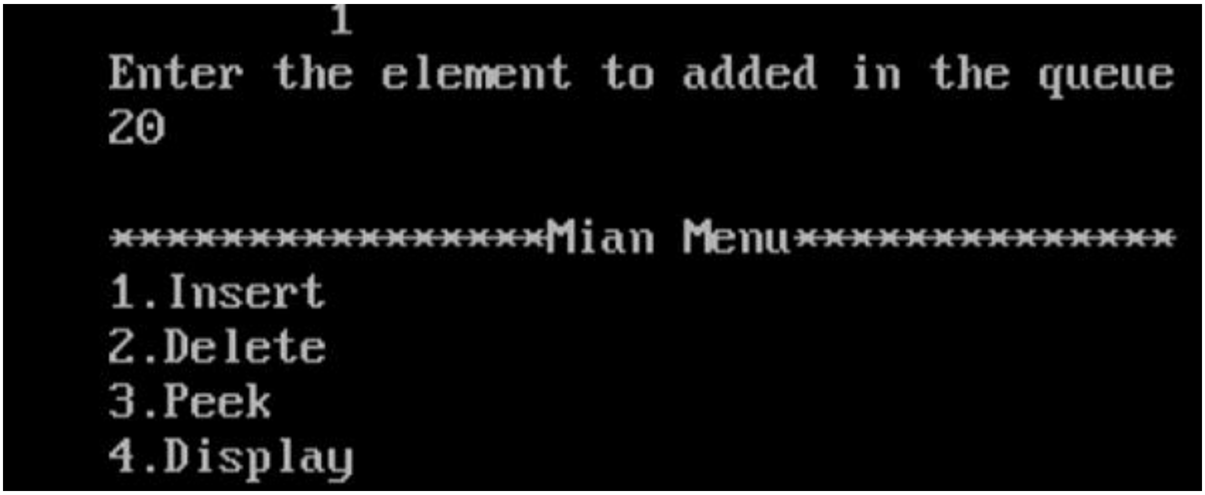
```
            case 1:
                printf("Enter the element to added in the queue\n");
                scanf("%d",&val);
                q=insert(q,val);
                break;
            case 2:
                q=delete_element(q);
                break;
            case 3:
                val=peek(q);
                if(val!=-1)
                    printf("The value at the front of queue is %d\n",val);
                break;
            case 4:
                q=display(q);
                break;
        }
    }while(ch>=1&&ch<=4);
    getch();
}
void create_queue(struct queue *q)
{
    q->rear=NULL;
    q->front=NULL;
}
struct queue *insert(struct queue *q,int val)
{
    struct node *ptr;
    ptr=(struct node *)malloc(sizeof(struct node));
    ptr->data=val;
    if(q->front==NULL)
    {
        q->front=ptr;
        q->rear=ptr;
        q->front->next=NULL;
        q->rear->next=NULL;
    }
    else
    {
        q->rear->next=ptr;
        q->rear=ptr;
        q->rear->next=NULL;
    }
    return q;
}
struct queue *display(struct queue *q)
{
    struct node *ptr;
    ptr=q->front;
    if(ptr==NULL)
        printf("Queue is empty\n");
    else
    {
        printf("\n");
        while(ptr!=q->rear)
        {
            printf("%d\t",ptr->data);
            ptr=ptr->next;
        }
        printf("%d\t",ptr->data);
    }
    return q;
```

```
}
struct queue *delete_element(struct queue *q)
{
        struct node *ptr;
        ptr=q->front;
        if(q->front==NULL)
                printf("Underflow\n");
        else
        {
                q->front=q->front->next;
                printf("The value being deleted is %d\n",ptr->data);
                free(ptr);
        }
        return q;
}
int peek(struct queue *q)
{
        if(q->front==NULL)
        {
                printf("Queue is empty\n");
                return -1;
        }
        else
                return q->front->data;
}
```

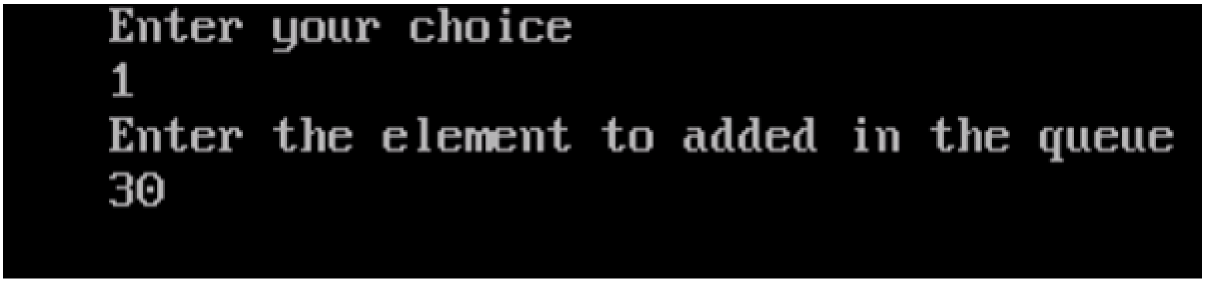
OUTPUT:



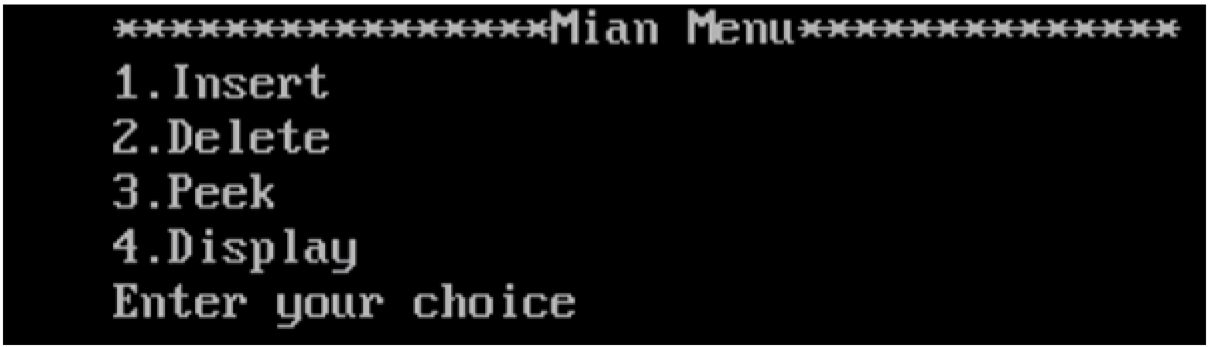
```
*****************Mian Menu***************
1.Insert
2.Delete
3.Peek
4.Display
Enter your choice
1
Enter the element to added in the queue
10

*****************Mian Menu***************
1.Insert
2.Delete
3.Peek
4.Display
Enter your choice
```



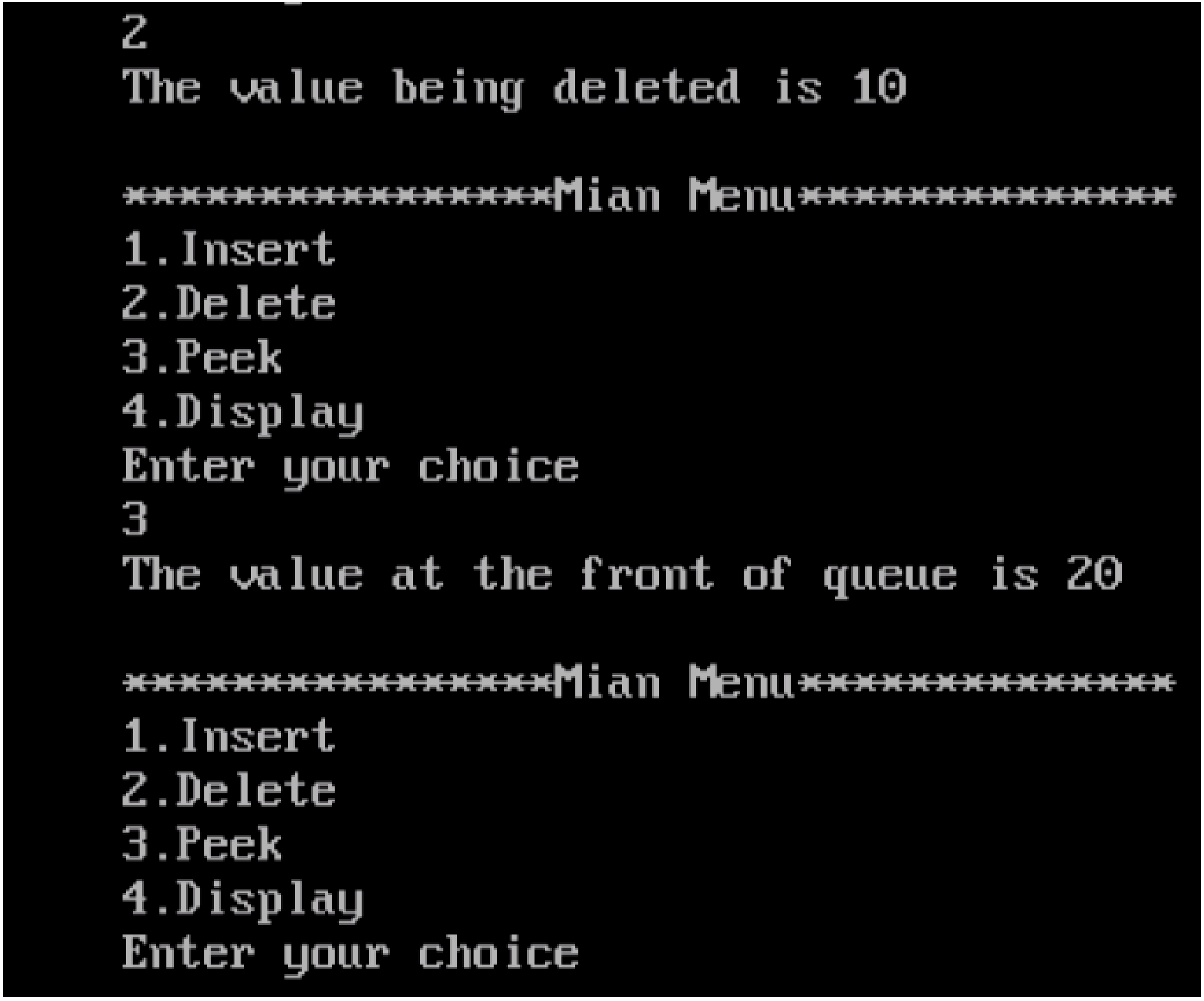
```
        1
Enter the element to added in the queue
20

*****************Mian Menu***************
1.Insert
2.Delete
3.Peek
4.Display
```



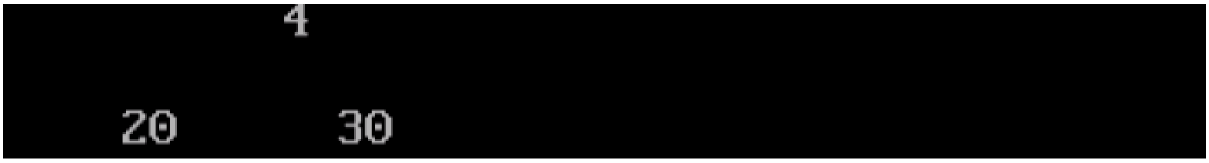
```
Enter your choice
1
Enter the element to added in the queue
30
```



```
****************Mian Menu**************
1.Insert
2.Delete
3.Peek
4.Display
Enter your choice
```



```
2
The value being deleted is 10

****************Mian Menu**************
1.Insert
2.Delete
3.Peek
4.Display
Enter your choice
3
The value at the front of queue is 20

****************Mian Menu**************
1.Insert
2.Delete
3.Peek
4.Display
Enter your choice
```



```
      4


20      30
```