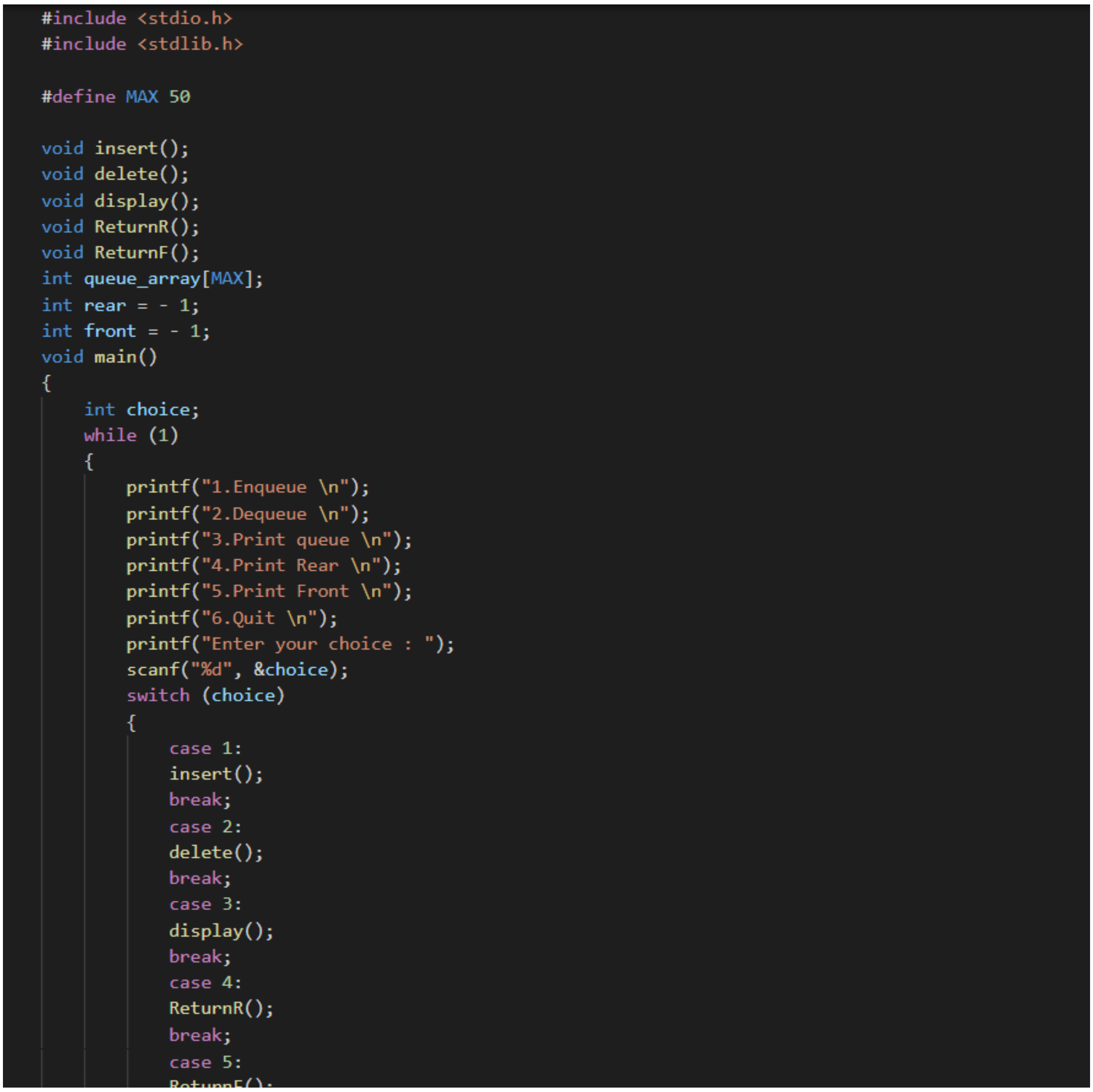
Agnim Gupta

2028083

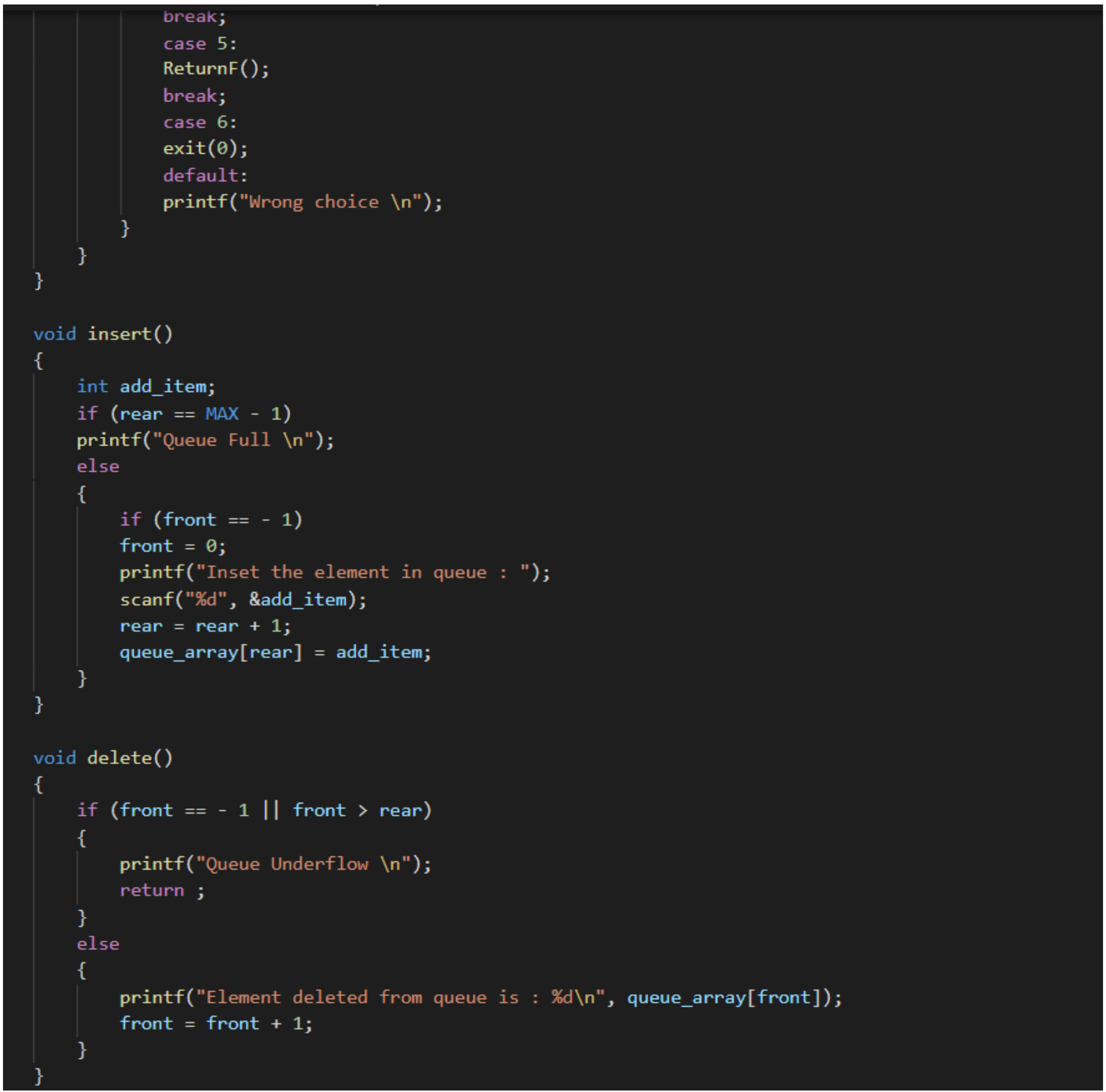
A23

CSSE

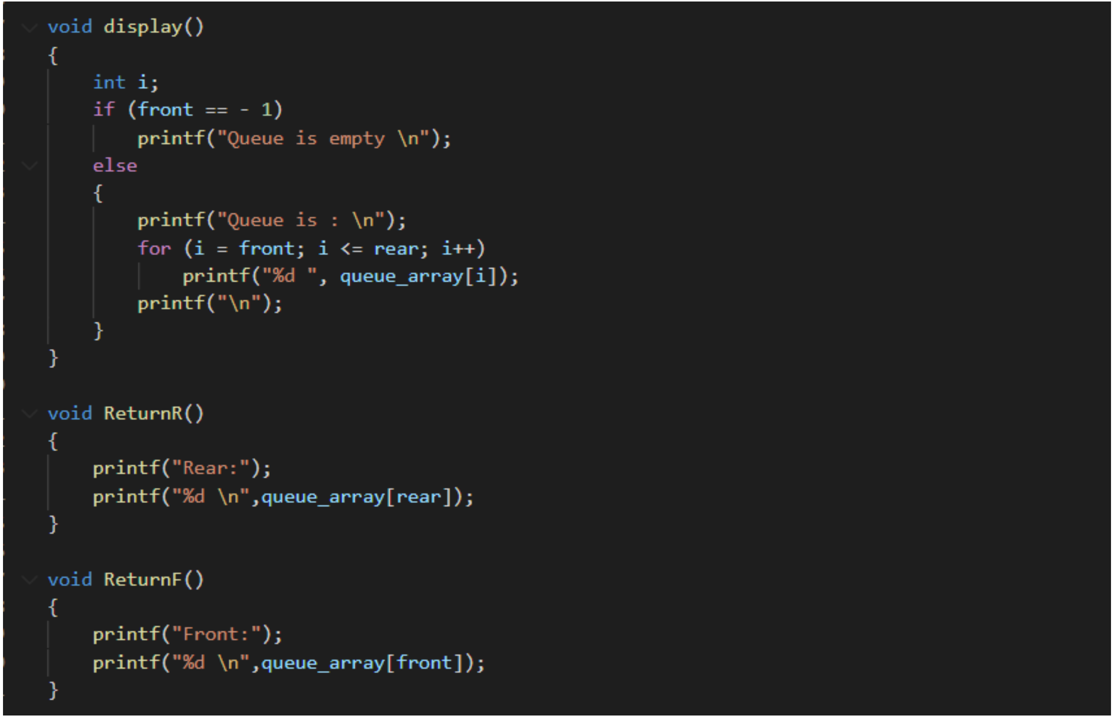
Question 1



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

#define MAX 50

void insert();
void delete();
void display();
void ReturnR();
void ReturnF();
int queue_array[MAX];
int rear = - 1;
int front = - 1;
void main()
{
    int choice;
    while (1)
    {
        printf("1.Enqueue \n");
        printf("2.Dequeue \n");
        printf("3.Print queue \n");
        printf("4.Print Rear \n");
        printf("5.Print Front \n");
        printf("6.Quit \n");
        printf("Enter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
            insert();
            break;
            case 2:
            delete();
            break;
            case 3:
            display();
            break;
            case 4:
            ReturnR();
            break;
            case 5:
            ReturnF();
```

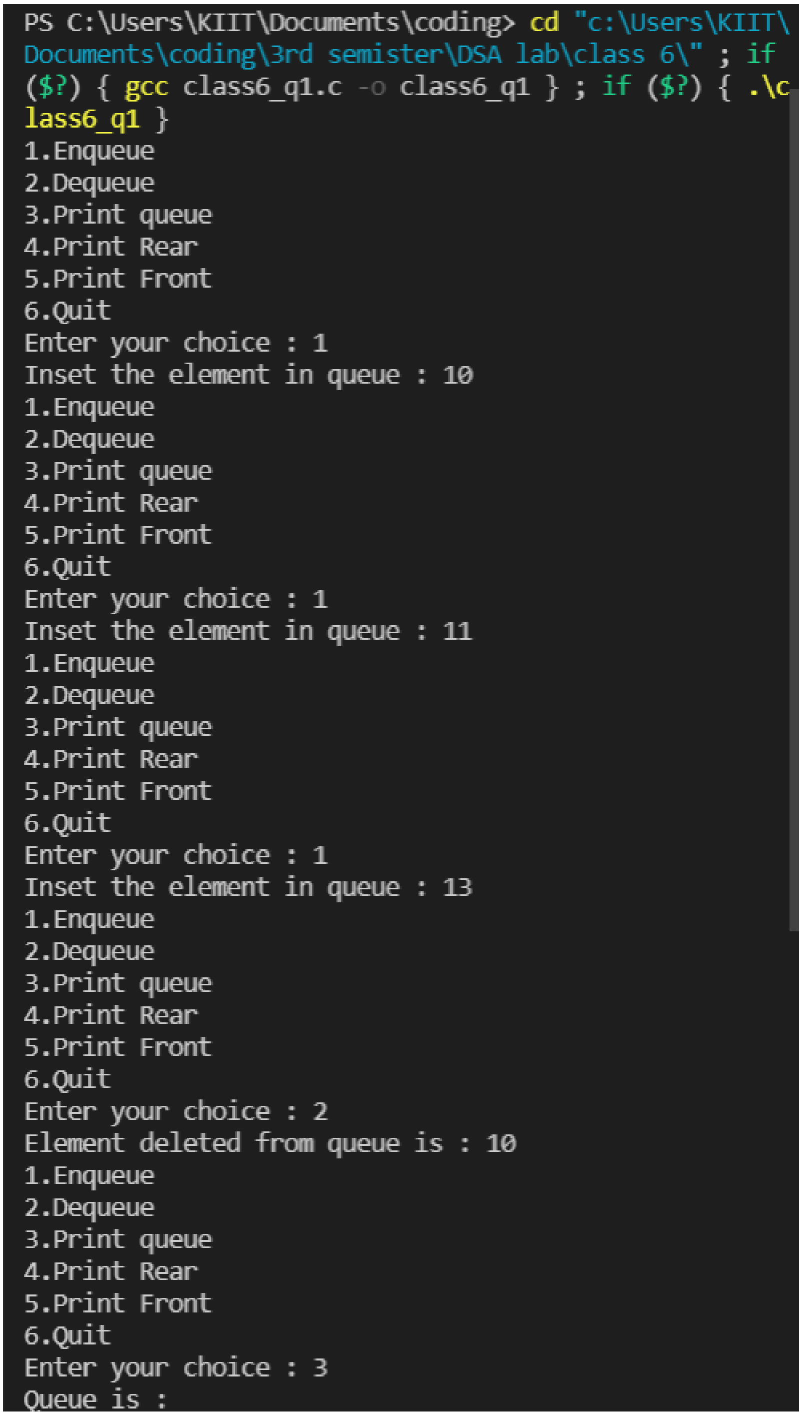


```
            break;
            case 5:
            ReturnF();
            break;
            case 6:
            exit(0);
            default:
            printf("Wrong choice \n");
        }
    }
}

void insert()
{
    int add_item;
    if (rear == MAX - 1)
    printf("Queue Full \n");
    else
    {
        if (front == - 1)
        front = 0;
        printf("Inset the element in queue : ");
        scanf("%d", &add_item);
        rear = rear + 1;
        queue_array[rear] = add_item;
    }
}

void delete()
{
    if (front == - 1 || front > rear)
    {
        printf("Queue Underflow \n");
        return ;
    }
    else
    {
        printf("Element deleted from queue is : %d\n", queue_array[front]);
        front = front + 1;
    }
}
```

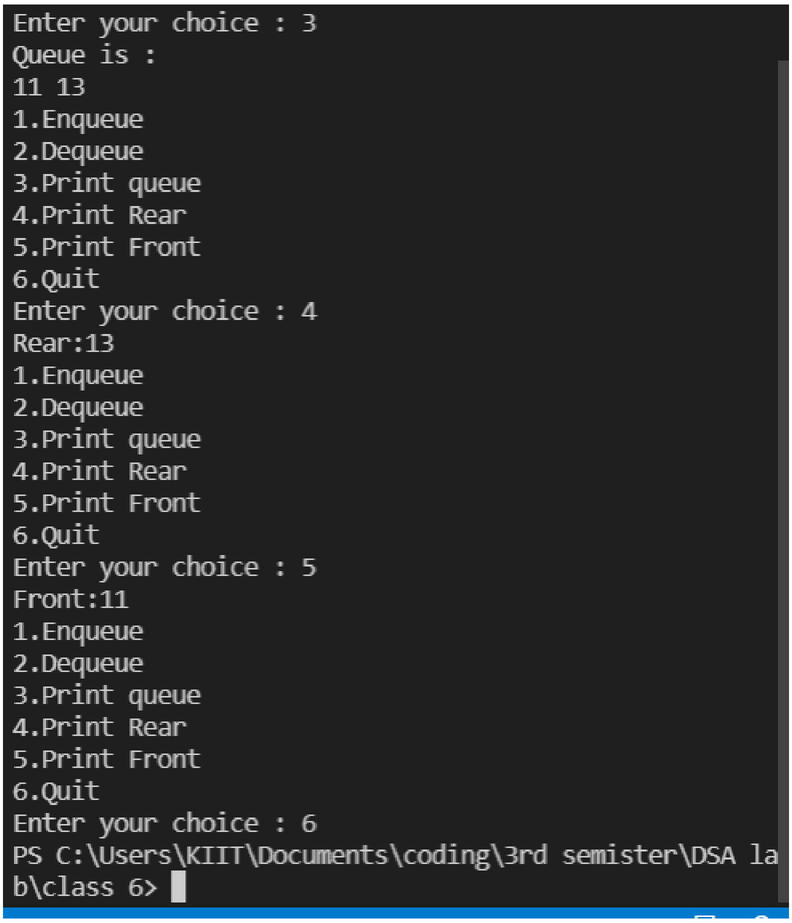


```
void display()
{
    int i;
    if (front == - 1)
        printf("Queue is empty \n");
    else
    {
        printf("Queue is : \n");
        for (i = front; i <= rear; i++)
            printf("%d ", queue_array[i]);
        printf("\n");
    }
}

void ReturnR()
{
    printf("Rear:");
    printf("%d \n",queue_array[rear]);
}

void ReturnF()
{
    printf("Front:");
    printf("%d \n",queue_array[front]);
}
```

Output

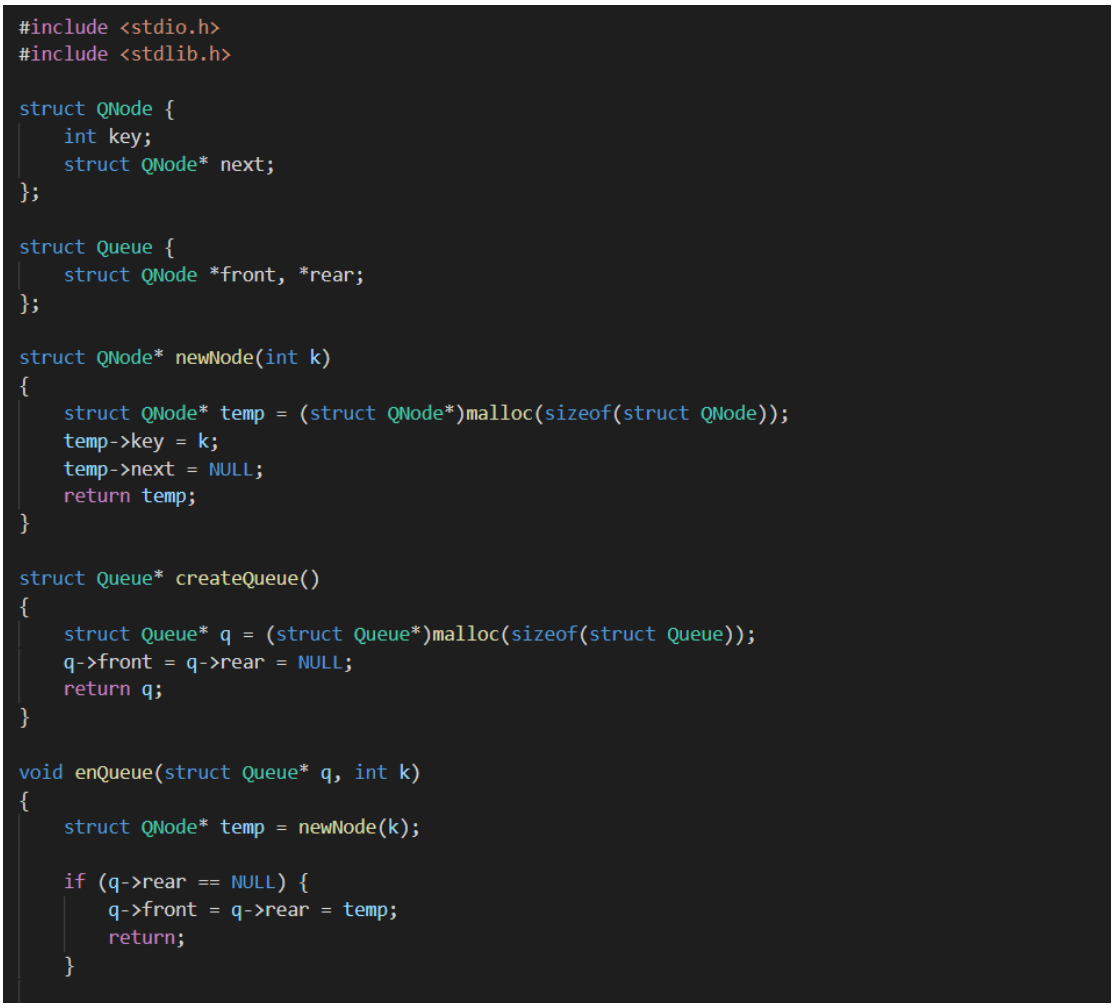


```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KIIT\
Documents\coding\3rd semister\DSA lab\class 6\" ; if
($?) { gcc class6_q1.c -o class6_q1 } ; if ($?) { .\c
lass6_q1 }
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 1
Inset the element in queue : 10
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 1
Inset the element in queue : 11
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 1
Inset the element in queue : 13
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 2
Element deleted from queue is : 10
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 3
Queue is :
```

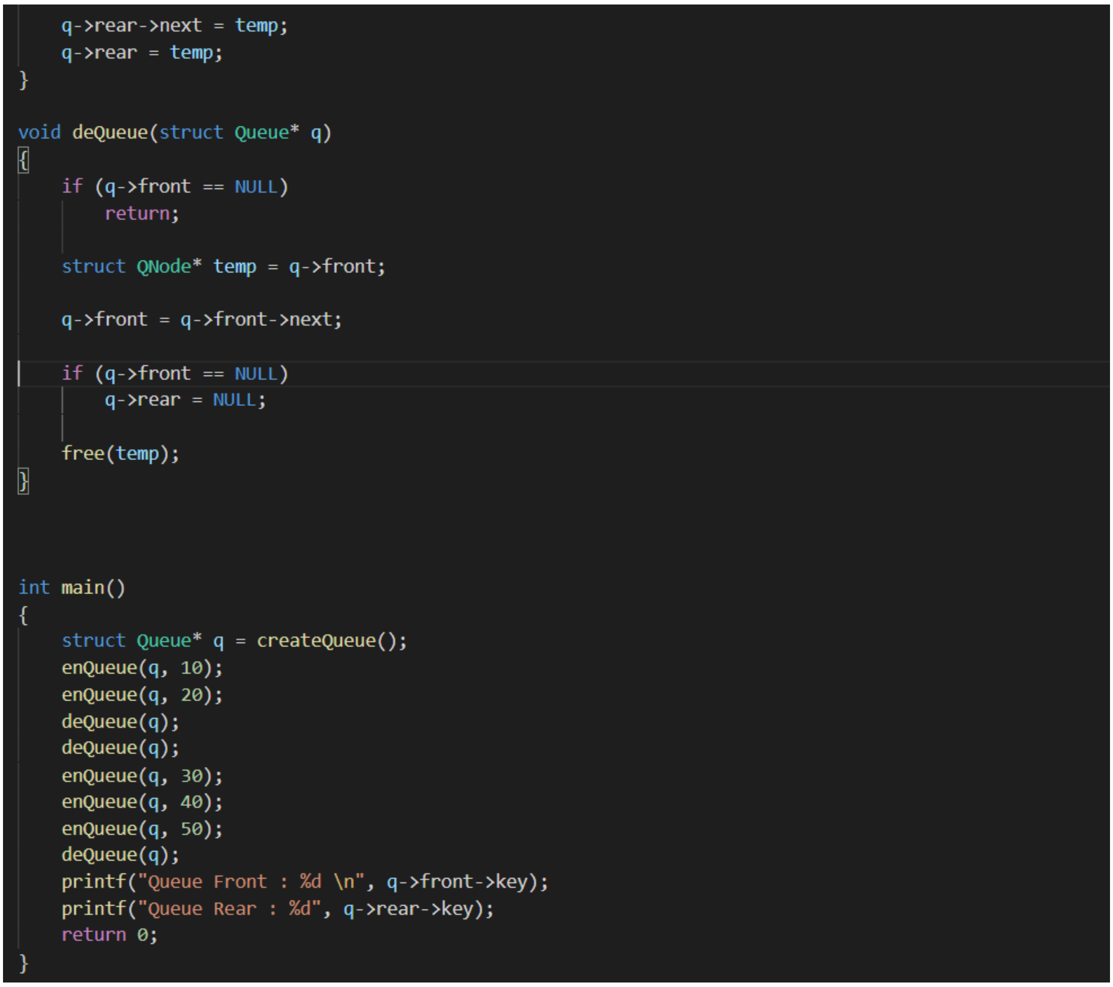


```
Enter your choice : 3
Queue is :
11 13
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 4
Rear:13
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 5
Front:11
1.Enqueue
2.Dequeue
3.Print queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 6
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA la
b\class 6>
```

Question 2

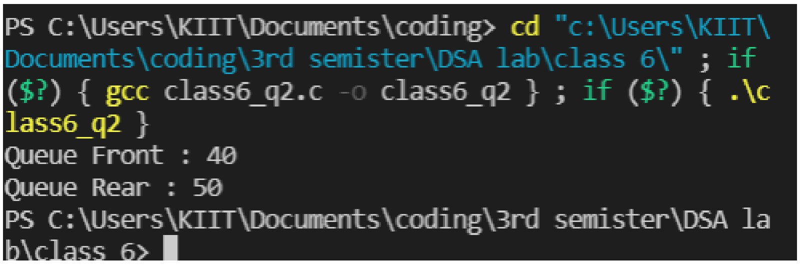


```c
#include <stdio.h>
#include <stdlib.h>

struct QNode {
    int key;
    struct QNode* next;
};

struct Queue {
    struct QNode *front, *rear;
};

struct QNode* newNode(int k)
{
    struct QNode* temp = (struct QNode*)malloc(sizeof(struct QNode));
    temp->key = k;
    temp->next = NULL;
    return temp;
}

struct Queue* createQueue()
{
    struct Queue* q = (struct Queue*)malloc(sizeof(struct Queue));
    q->front = q->rear = NULL;
    return q;
}

void enQueue(struct Queue* q, int k)
{
    struct QNode* temp = newNode(k);

    if (q->rear == NULL) {
        q->front = q->rear = temp;
        return;
    }
```



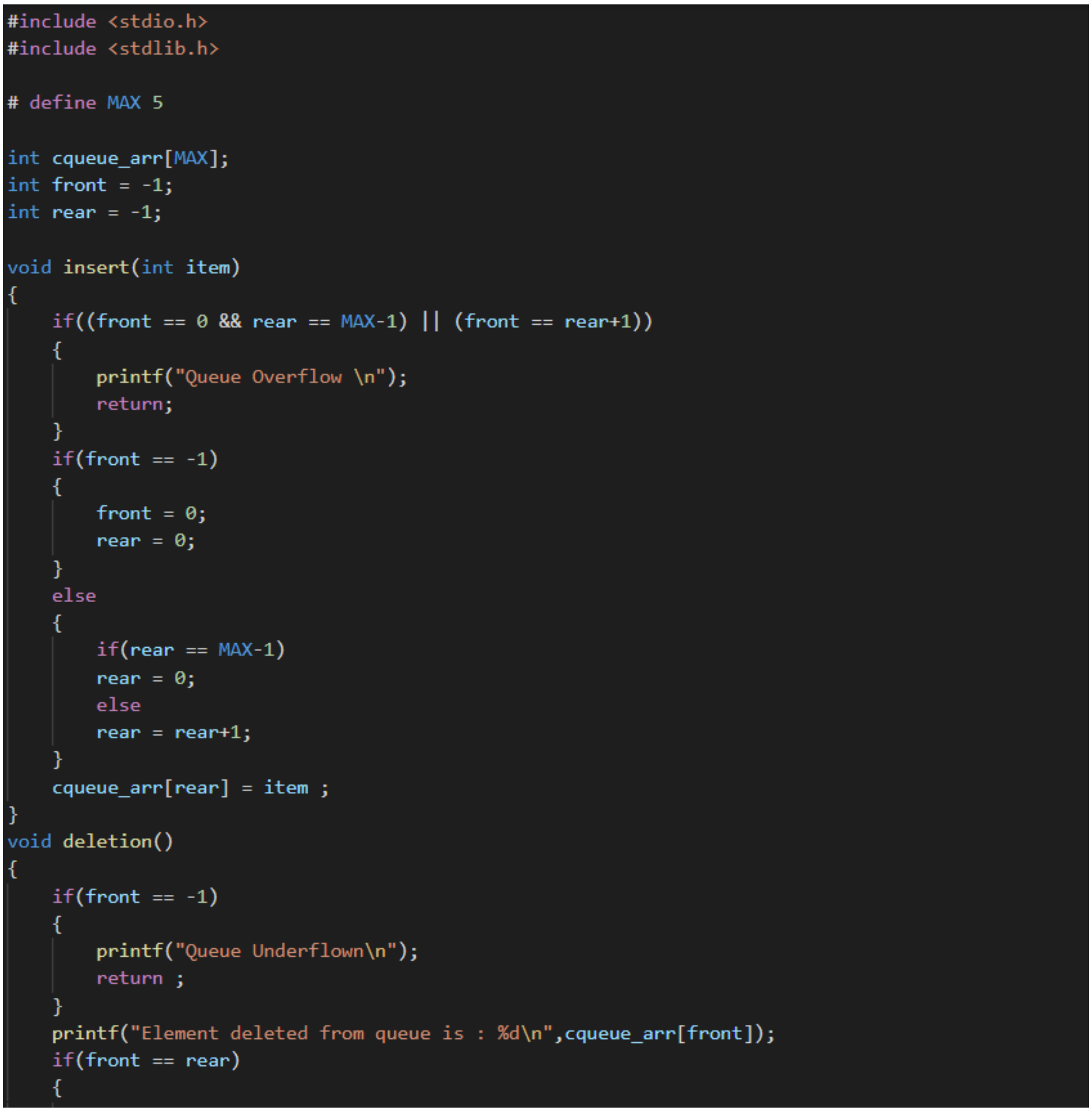
```c
    q->rear->next = temp;
    q->rear = temp;
}

void deQueue(struct Queue* q)
{
    if (q->front == NULL)
        return;

    struct QNode* temp = q->front;

    q->front = q->front->next;

    if (q->front == NULL)
        q->rear = NULL;

    free(temp);
}



int main()
{
    struct Queue* q = createQueue();
    enQueue(q, 10);
    enQueue(q, 20);
    deQueue(q);
    deQueue(q);
    enQueue(q, 30);
    enQueue(q, 40);
    enQueue(q, 50);
    deQueue(q);
    printf("Queue Front : %d \n", q->front->key);
    printf("Queue Rear : %d", q->rear->key);
    return 0;
}
```

Output

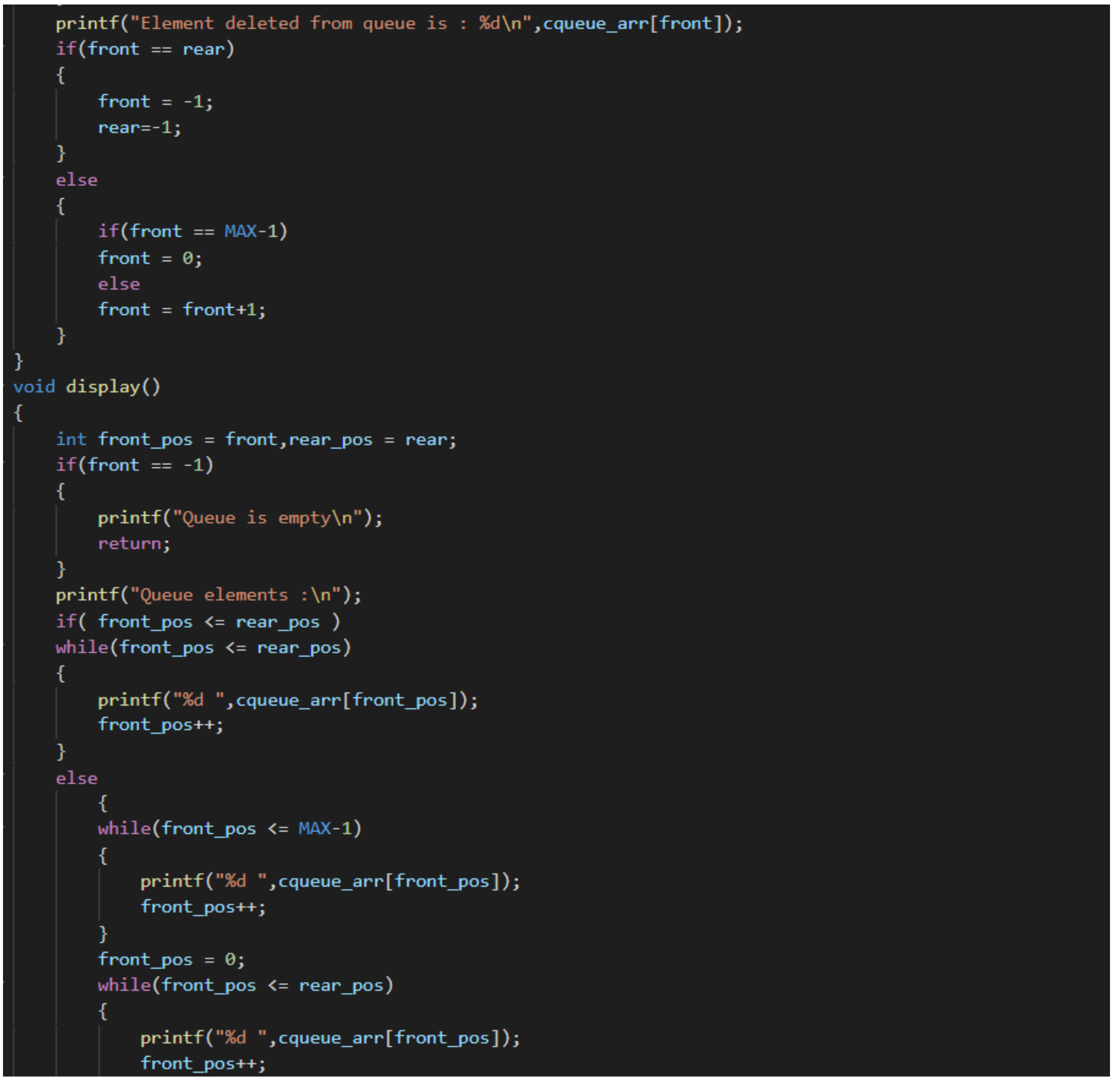


```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KIIT\
Documents\coding\3rd semister\DSA lab\class 6\" ; if
($?) { gcc class6_q2.c -o class6_q2 } ; if ($?) { .\c
lass6_q2 }
Queue Front : 40
Queue Rear : 50
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA la
b\class 6>
```

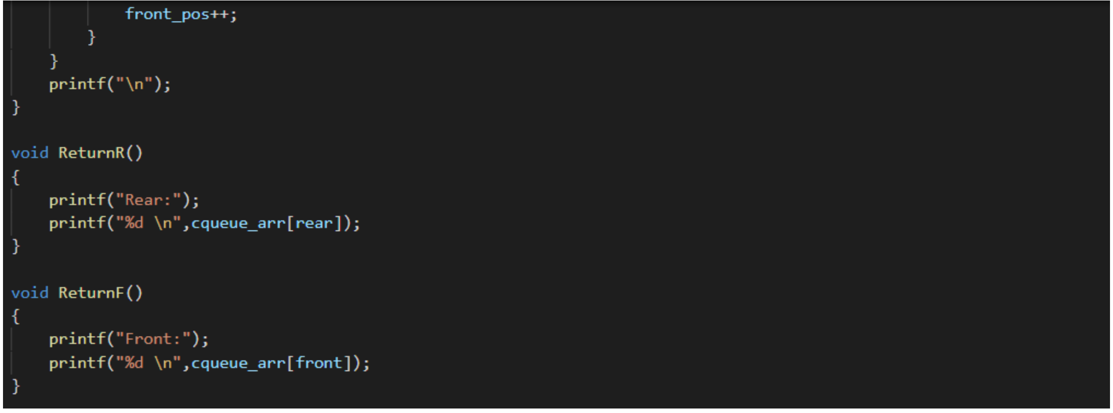
Question 3



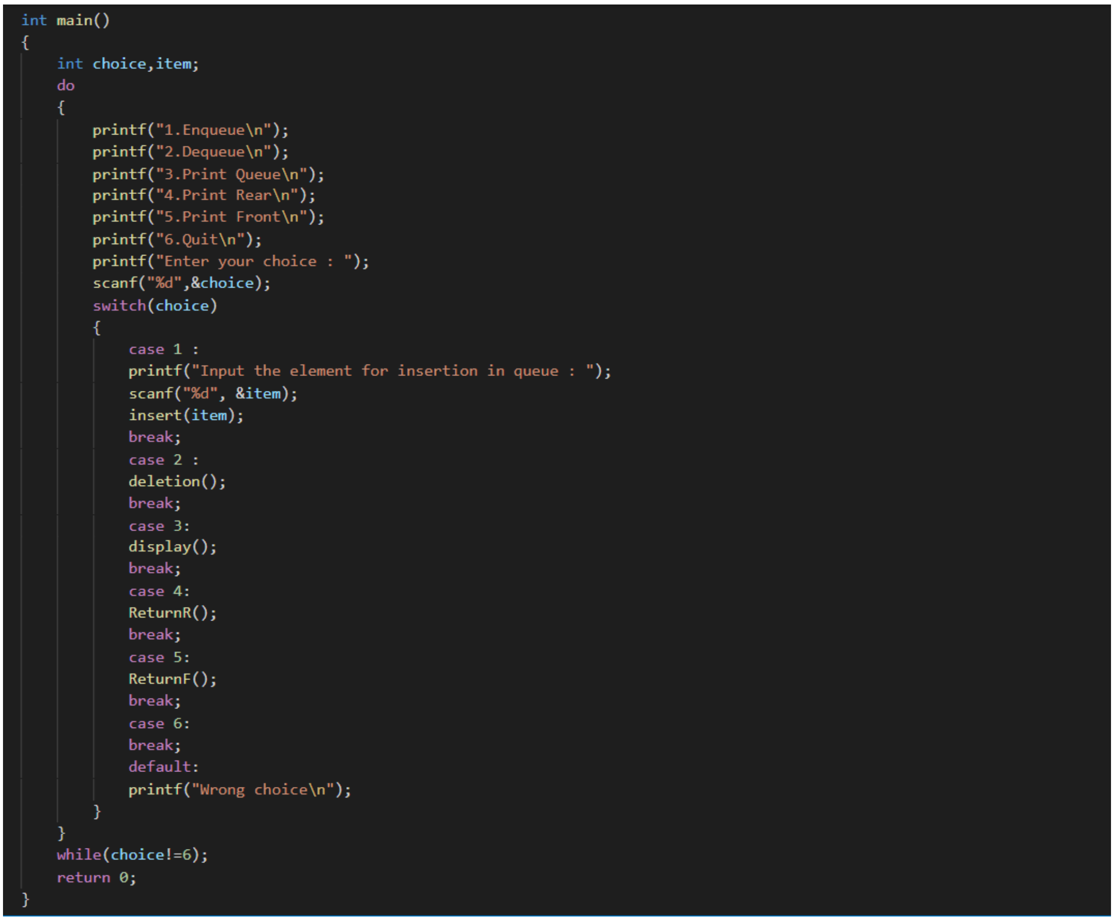
```c
#include <stdio.h>
#include <stdlib.h>

# define MAX 5

int cqueue_arr[MAX];
int front = -1;
int rear = -1;

void insert(int item)
{
    if((front == 0 && rear == MAX-1) || (front == rear+1))
    {
        printf("Queue Overflow \n");
        return;
    }
    if(front == -1)
    {
        front = 0;
        rear = 0;
    }
    else
    {
        if(rear == MAX-1)
        rear = 0;
        else
        rear = rear+1;
    }
    cqueue_arr[rear] = item ;
}
void deletion()
{
    if(front == -1)
    {
        printf("Queue Underflown\n");
        return ;
    }
    printf("Element deleted from queue is : %d\n",cqueue_arr[front]);
    if(front == rear)
    {
```



```
    printf("Element deleted from queue is : %d\n",cqueue_arr[front]);
    if(front == rear)
    {
        front = -1;
        rear=-1;
    }
    else
    {
        if(front == MAX-1)
        front = 0;
        else
        front = front+1;
    }
}
void display()
{
    int front_pos = front,rear_pos = rear;
    if(front == -1)
    {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue elements :\n");
    if( front_pos <= rear_pos )
    while(front_pos <= rear_pos)
    {
        printf("%d ",cqueue_arr[front_pos]);
        front_pos++;
    }
    else
        {
        while(front_pos <= MAX-1)
        {
            printf("%d ",cqueue_arr[front_pos]);
            front_pos++;
        }
        front_pos = 0;
        while(front_pos <= rear_pos)
        {
            printf("%d ",cqueue_arr[front_pos]);
            front_pos++;
```

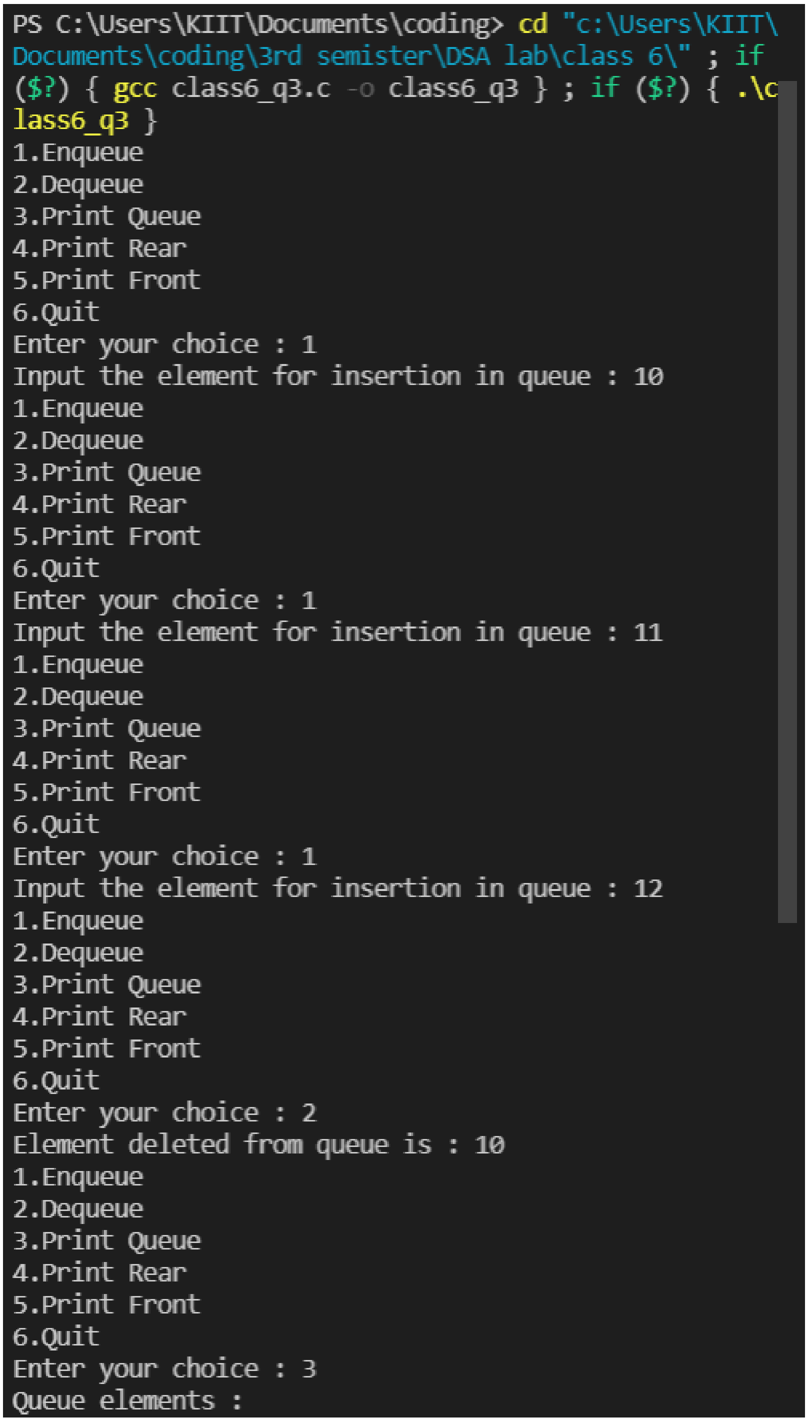


```
            front_pos++;
        }
    }
    printf("\n");
}

void ReturnR()
{
    printf("Rear:");
    printf("%d \n",cqueue_arr[rear]);
}

void ReturnF()
{
    printf("Front:");
    printf("%d \n",cqueue_arr[front]);
}
```

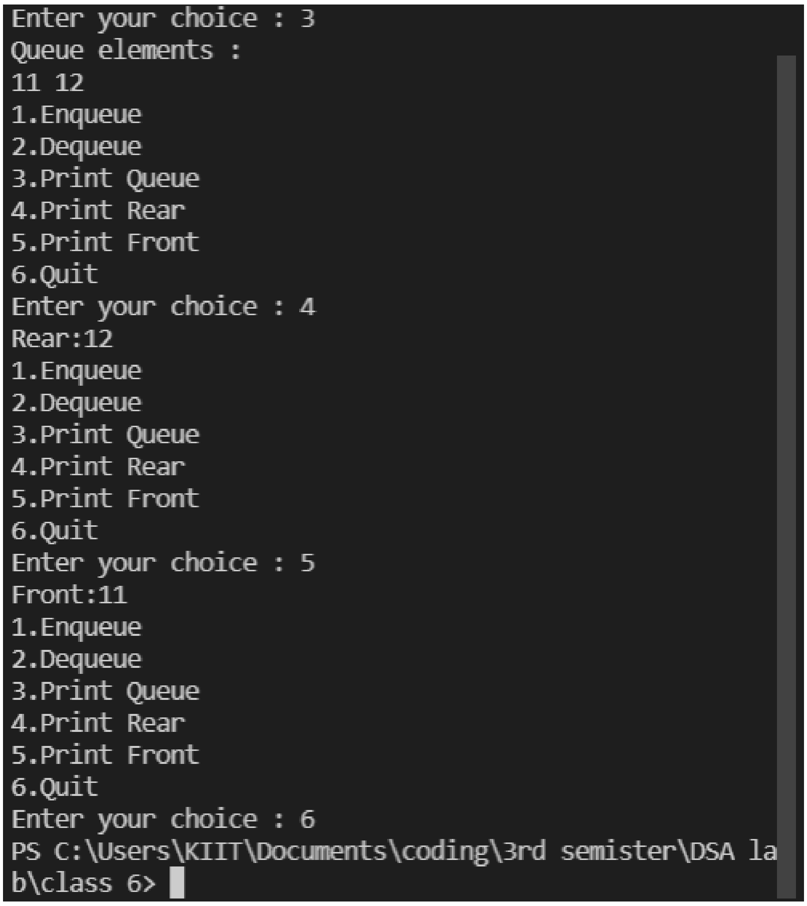


```c
int main()
{
    int choice,item;
    do
    {
        printf("1.Enqueue\n");
        printf("2.Dequeue\n");
        printf("3.Print Queue\n");
        printf("4.Print Rear\n");
        printf("5.Print Front\n");
        printf("6.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 :
            printf("Input the element for insertion in queue : ");
            scanf("%d", &item);
            insert(item);
            break;
            case 2 :
            deletion();
            break;
            case 3:
            display();
            break;
            case 4:
            ReturnR();
            break;
            case 5:
            ReturnF();
            break;
            case 6:
            break;
            default:
            printf("Wrong choice\n");
        }
    }
    while(choice!=6);
    return 0;
}
```

Output



```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KIIT\
Documents\coding\3rd semister\DSA lab\class 6\" ; if
($?) { gcc class6_q3.c -o class6_q3 } ; if ($?) { .\c
lass6_q3 }
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 1
Input the element for insertion in queue : 10
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 1
Input the element for insertion in queue : 11
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 1
Input the element for insertion in queue : 12
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 2
Element deleted from queue is : 10
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 3
Queue elements :
```



```
Enter your choice : 3
Queue elements :
11 12
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 4
Rear:12
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 5
Front:11
1.Enqueue
2.Dequeue
3.Print Queue
4.Print Rear
5.Print Front
6.Quit
Enter your choice : 6
PS C:\Users\KIIT\Documents\coding\3rd semister\DSA la
b\class 6>
```