

```

#include <stdio.h>

# define max 6
int queue[max];
int front=-1;
int rear=-1;
void enqueue(int element)
{
    if(front== -1 && rear== -1)
    {
        front=0;
        rear=0;
        queue[rear]=element;
    }
    else if((rear+1)%max==front)
    {
        printf("Queue is overflow..");
    }
    else
    {
        rear=(rear+1)%max;
        queue[rear]=element;
    }
}

int dequeue()
{
    if((front== -1) && (rear== -1))
    {
        printf("\nQueue is underflow..");
    }
    else if(front==rear)
    {
        printf("\nThe dequeued element is %d", queue[front]);
        front=-1;
        rear=-1;
    }
    else
    {
        printf("\nThe dequeued element is %d", queue[front]);
        front=(front+1)%max;
    }
}

void display()
{
    int i=front;
    if(front== -1 && rear== -1)

```

```

    {
        printf("\n Queue is empty..");
    }
    else
    {
        printf("\nElements in a Queue are :");
        while(i<=rear)
        {
            printf("%d,", queue[i]);
            i=(i+1)%max;
        }
    }
}

void search()
{
    int item,i,c=0;
    printf("Enter the element which is to be searched");
    scanf("%d", &item);
    for(i=front;i<=rear;i++)
    {

        if(item==queue[i])
        {
            printf("item found at location %d ",i+1);
            c++;
        }
    }
    if(c==0)
        printf("item not found");
    else
        printf("item found at location %d times in the queue ",c);
}

int main()
{
    int choice=1,x;    // variables declaration

    while(choice<4 && choice!=0)    // while loop
    {
        printf("\n Press 1: Insert an element");
        printf("\nPress 2: Delete an element");
        printf("\nPress 3: Display the element");
        printf("\nPress 4: search the element");
        printf("\nEnter your choice");
        scanf("%d", &choice);

        switch(choice)
        {

```

```
    case 1:

        printf("Enter the element which is to be inserted");
        scanf("%d", &x);
        enqueue(x);
        break;
    case 2:
        dequeue();
        break;
    case 3:
        display();
        break;3

    case 4:
        search();

}}
return 0;
}
```