16/10/2020

1BM19CS053
Divyendu Jhanwar

```
#define size 5
int  Queue[size] ;
int   front = -1;
int   rear = -1;

Void  enqueue(int x)
{
      if (rear== size -1)
      {
            printf ("Queue is full \n"),
      }
      else if (front==-1 && rear==-1)
      {
            front ++;
            rear ++;
            Queue[rear]= x;
      }
      else
      {
            rear++
            Queue[rear]= x;
      }
}

int  Dequeue()
{
      int x;
      if (front == -1)
      {
```

1BM19CS057
Divyanshu Thanwar

```
        return -1;
    }
    else
    {
        x = Queue[front];
        front++;
        if(front > rear)
        {
            front = -1;
            rear = -1;
        }
        return x;
    }
}

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

1BM19CS053
Divyendu Jhonwar

```
int main()
{

    int i;
    int x;
    do {
        printf("\n 1. Insert to Queue");
        printf("\n 2. Delete from the Queue");
        printf("\n 3. Display the content ");
        printf(" \n4. Exit\n");
        printf("Enter the option : ");
        ~~switch (i)~~
        scanf("%d", & i);
        switch (i)
        {
            case 1 : printf(" Enter the element \n");
                     scanf("%d", & x);
                     enqueue(x);
                     break;
            case 2 : x = ~~deq~~ Dequeue()
                      if(x ==-1)
                        printf("Queue is empty \n");
                     else
                        printf(" Removed element from the queue
                                %d ", x);
                     break;
            case 3 : display();
                     break;
            case 4 : break;
        }
```

1BM19CS053
Divyanshu Thanwar

```
} while (i != 4)
return 0;
}
```