LAB-6

1] Circular queue:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
int front =-1;
int rear = -1;

int queue [MAX];
void Enqueue(int);
int Dequeue();
void display();
int main()
{
    int option;
    int item;
    do {
        printf(" Circular Queue \n");
        printf("\n1. Insert to Queue (EnQueue)");
        printf("\n. delete from the Queue (DeQueue)");
        printf("\n Display the content ");
        printf("\n 4. Exit \n");
        printf(" Enter the option : ");
scanf("%d", &option);
switch(option)
{
    case 1: printf("Enter the element \n");
            scanf("%d", & item);
            Enque(item);
            break;
    case 2: item = Deque();
            if (item == -1)
               printf("Queue is empty \n");
            else
               printf(" Removed element from
                         the que %d", item);

               break;
    case 3: display();
            break;
    case 4: exit (0);
  }
} while (option != 4);
return 0;
}
```

Scanned by TapScanner

```
void Enque( int ele)
{
    if ((rear+1) % MAX == front)
    printf("Queue is full\n");
    else
    {
        rear = (rear+1) % MAX;
        queue [rear] = ele;
        if (front == -1)
          front = 0;
     }
}
int Deque()
{
    int item;
    if ((front == -1) && (rear == -1))
      return -1;
    else
    {
       item = queue [front];
       front = (front +1) % MAX;
       if (front > rear)
       {
          front = -1;
          rear = -1;
       }
       return item;
```

Scanned by TapScanner

```
    }
}
void display()
{
    int i ;
    if ((front == 0) && (rear==-1))
     printf ("Queue is empty \n");
    else
    {
       printf("\n Queue contents : ");
       for (i = front ; i <= rear; i++)
            printf("%d ", queue[i]);
     } printf("\n");
}
```

expected output :

```
Circular Queue
1. Insert to Queue (Enqueue)
2. delete from the Queue (Dequeue)
3. Display the Content
4. Exit
Enter the option : 2
Queue is empty
```

Scanned by TapScanner

```
Circular Queue

1. Insert to Queue (EnQueue)

2. Delete from the Queue (Dequeue)

3.   Display the content

4. Exit

Enter the option : 1.

 Enter the element

  2.

Circular Queue.

 1. Insert to Queue (EnQueue)

 2. delete from the Queue (DeQueue)

 3. Display the content

 4. exit.

Enter the option : 4.
```

Scanned by TapScanner

Priority Queue.

```
#include <stdio.h>
#include <stdlib.h>
#define N 3

int queue [3] [N];
int front [3] = {0, 0, 0};
int rear [3] = {-1, -1, -1};
int item, pr ;
void pq insert (int pr)
{
    if (rear [pr] == N-1)
      printf ("\n Que overflow \n");
    else
    {
        printf ("\nenter the item \n");
        scanf ("%d", &item);
        rear [pr]++;
        queue [pr] [rear [pr]] = item ;
    }
    return;
}
void pqdelete ()
{   int i ;
    for (i=0 ; i<3 ; i++)
```

Scanned by TapScanner

```
{
    if (rear[i] == front[i] -1)
     printf("\n queue %d empty \n", i+1);
    else
    {
     printf("\n deleted item is %d of queue %d \n",
                              queue[i][front[i]],
     front[i]++;
     return;
     }
    }
  

}
void display()
{
   int i, j;
   for(i=0 ; i<3; i++)
   {
     if (rear[i] == front[i] -1)
      printf("\n queue %d empty \n", i+1);
     else
     {
      printf(" \n QUEUE %d: ", i+1);
      for (j = front[i]; j<= rear[i]; j++)
       printf("%d\t", queue[i] [j]);
```

Scanned by TapScanner

```
    };
    return;
}
int main()
{
    int ch;
    while(1)
    {
        printf("\n\t 1: PQ insert \n");
        printf("\n\t 2: PQ delete \n");
        printf("\n\t 3: PQ display \n");
        printf("\n\t 4: Exit \n");
        printf("\n Enter the choice\n");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1 : printf("\n enter the priority number\n");
                     scanf("%d", &pr);
                     if (pr>0 && pr<4)
                       pq insert(pr-1);
                     else
                     printf("\n only 3 priority exists 1 2 3\n");
                      break;
            case 2 : pq deletete();
                     break;
```

Scanned by TapScanner

```
        case 3 : display();
                 break;
        case 4 : exit(0);
        }
    }
    return 0;
}
```

Expected Output.

```
    1: PQ insert
    2: PQdelete
    3: PQ display
    4: Exit
enter the choice
1
enter the priority number
2
enter the item
45
        1: PQ insert
        2: PQdelete
        3: PQdisplay
        4: Exit
```

Scanned by TapScanner

```
enter the choice
1
enter the priority number
1
enter the item
67
    1: PQ insert
    2: PQ delete
    3: PQ display
     4: Exit
enter the choice
3
QUEUE 1: 67
QUEUE 2: 45
queue 3 empty
        1: PQ insert
        2: PQ delete
        3: PQ display
         4: Exit
 enter the choice
2
deleted item  is 67 of queue 1
```
Scanned by TapScanner

```
1: PQ insert
2: PQ delete
3: PQdisplay
4: Exit
enter the choice
4
```

Scanned by TapScanner