CIRCULAR QUEUE

```c
#include <stdio.h>
#include <stdlib.h>
#define qsize 5
int item, f=0, r=-1, q[qsize], count=0, ch;
void insertrear()
{
    if (count==qsize)
    {
        printf("Queue overflow \n");
        return;
    }
    r=(r+1)%qsize;
    q[r] = item;
    count++;
}
int deletefront()
{
    if (count==0) return -1;
    item =q[f];
    f=(f+1)%qsize;
    count = count-1;
    return item;
}
void displayq()
{
```

```c
    int i, front;
    if (count == 0)
    {
        printf(" Queue is empty \n");
        return;
    }

    front = f;
    printf(" Contents of queue \n");
    for (i=1; i<=count; i++)
    {
        printf ("%d\n", q[front]);
        front = (front+1) % qsize;
    }
}

int main ()
{
    for (;;)
    {
        printf ("\n1.insert_rear \n 2. delete_front \n3. display \n 4. exit \n");
        printf (" Enter choice \n");
        scanf ("%d", &ch);
        switch (ch)
        {
            case 1: printf(" enter the item to be inserted \n");
                scanf ("%d", &item);
                insertrear();
                break;
```

Expt. No. _____

```
case 2 : item=deletefront();
          if (item == -1)
          printf (" Queue is empty \n");
          else

          printf (" Item deleted = %d \n", item);
          break;
   case 3 : display();
            break;
default :  exit (0);
          }
      }

   return 0;
   }
```

LINEAR QUEUE

```c
#include <stdio.h>
#include <stdlib.h>
#define qsize 5
int item, front=0, rear=-1, q[10];
void insert rear()
{
    if (rear == qsize-1)
    {
    printf("Queue overflow|n");
     return;
    }
    rear = rear+1;
    q[rear] = item;
}
int delete front()
{
    if (front > rear)
    {
        front = 0;
        rear = -1;
        return -1;
    }
    return q[front++];
}
void display q()
```

```c
{
    int i;
    if (front> rear)
    {
        printf(" Queue is empty \n");
        return;
    }
    printf(" Contents of queue \n");
    for ( i = front; i< = rear; i++)
    {
        printf ("%d \n", q [i]);
    }
}

int main()
{
    int ch;
    for (;;)
    {
        printf (" \n1. insert_rear\n2. delete_front_\n3.display \n4.exit \n");
        printf (" enter choice\n");
        scanf ("%d", &ch);
        switch (ch)
        {
            case1: printf ("enter the item to be inserted \n");
                   scanf ("%d, & item);
                   insert_rear ();
                   break;
```

```
case 2 :   item = deletefront ();
           if (item == -1)
           printf (" Item Queue is empty \n");
           else
           printf (" Item deleted = %d \n", item);
           break;
case 3 :   display Q();
           break;
default : exit (0);
           }
           }

return 0;
}
```