Name : Sakshi.P.Khandoba
Sem : 3rd Section : C
Bath : 2
USN : 1BM19CS139

papergrid
Date: 19/10/20

LAB PROGRAM

Linear Queue :

```
#include <stdio.h>
#include <stdlib.h>
# define queue_size 3
int item, front = 0, rear = -1, q[10];
void insertrear()
{
    if (rear == queue_size - 1)
    {
        printf("Queue Overflow. \n");
        return;
    }
    rear = rear + 1;
    q[rear] = item;
}
int delfront()
{
    if (front > rear)
    {
        front = 0;
        rear = -1;
        return -1;
    }
    return q[front++];
}
void display()
{
    int i;
    if (front > rear)
    {
```

```
        printf("Queue is empty. \n");
        return;
    }
    printf("Contents of the Queue are: \n");
    for(i=front; i<=rear; i++)
    {
        printf("%d \n", q[i]);
    }
}
int main()
{
    int choice;
    for(;;)
    {
        printf("\n1: Insert rear \n2: Delete front
        \n3: Display \n4: Exit\n");
        printf("Enter the choice:");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: printf("Enter item to be inserted:\n");
            scanf("%d", &item);
            insertrear();
            break;
            case 2: item = delfront();
            if(item == -1)
                printf("Queue is empty. \n");
            else
                printf("Item deleted is %d \n", item);
            break;
            case 3: display();
            break;
            default: exit(0);
        }
    }
}
```

Name : Sakshi. P. Khandoba
Sem : 3rd Section : C
Batch: 2
USN : 1BM19CS139

papergrid
Date: 19/10/20

LAB PROGRAM

Circular Queue :

```
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
#define queue_size 3
int item, front = 0, rear = -1, q[queue_size], count=0;
void insertrear()
{
    if(count == queue_size)
    {
        printf("Queue overflow.");
        return;
    }
    rear = (rear +1)% queue_size;
    q[rear] = item;
    count++;
}
int deletefront()
{
    if(count == 0)
        return -1;
    item = q[front];
    front = (front+1) % queue_size;
    count = count - 1;
    return item;
}
void display()
{
    int i,f;
    if (count == 0)
```

```
{
    printf("The queue is empty.");
    return;
}
f = front;
printf("Contents of the queue are : \n");
for(i=0; i<= count; i++)
{
    printf("%d \n", q[f]);
    f = (f+1) % queue_size;
}
}
void main()
{
  int choice;
  for(;;)
  {
    printf("\n 1. Insert rear \n2. Delete front
    \n3. Display \n4. Exit \n");
    printf("Enter the choice :");
    scanf("%d", &choice);
    switch (choice)
    {
        case 1 : printf("Enter item to be inserted:");
                 scanf("%d", &item);
                 insertrear();
                 break;
        case 2 : item = deletefront();
                 if (item == -1)
                    printf("Queue is empty. \n");
                 else
                    printf("Item deleted is %d. \n",
                     item);
                 break;
```

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