papergrid

Date: / /

LAB3 Linear Queue

1)
```
#include<stdio.h>
#define qsize 5
int item, front=-1, 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 q[front++];
  }
void display_q()
{
  int i;
  if (front>rear)
  printf ("Queue empty \n");
   else
   for ( i=front; i<= rear; i++)
     {
      printf ("%d \n", q[i]);
    }
  }
```

papergrid

Date: / /

```
void main()
{
 int choice;
 for( ; ; )
{
 printf ("\n1.insertrear\n2.deletefront\n3.
          display\n4.exit\n");
 printf ("enter the choice\n");
 scanf("%d", &choice);
 switch(choice)
  {
   case1: printf ("enter the item to be inserted\n");
          scanf ("%d", &item);
          insert_rear();
          break;
   case2:
          item=delete_front();
          if (item ==-1)
          printf ("empty queue\n");
          else
          printf ("element deleted:", item);
          break;
   case3:
          display_Q();
          break;
   default:
        printf ("end of operation\n");
   }
  }
 }
```

2) circular Queue

```
#include<stdio.h>
#include <stdlib.h>
# define q-size 5
int item, front =0, rear =-1, q [q-size], count=0
void insert-rear()
 {
  if (count==q-size)
  {
   printf ("queue underflow \n");
  return;
  }
  rear=(rear+1) % q-size;
  q[rear]=item;
  count ++;
}
 int deletefront()
{
  if (count==0)
  return -1;
 item = q[front];
 front =(front+1) % q-size;
  count --;
  return item;
}
  void displayq()
 {
      int i ,f;
      if (count==0)
     {
        printf("queue is empty \n");
      return;
    }
f=front;
printf("contents of the queue\n");
for(i=1; i<=count; i++)
 {
   printf("%d\n", q[f]);
   f=(f+1)%q_size;
 }
}
void main()
{
int choice;
for(;;)
{
printf("\n1.insertrear\n2.deletefront\n3.diplay
          \n4.exit\n");
scanf("%d",&choice);
switch(choice)
{
 case 1: printf("enter the item to be inserted\n");
         scanf("%d",&item);
         insert_rear();
         break;
 case 2 : item=delete_front();
         if (item==-1)
         printf("\n empty queue");
         else
         printf("item deleted %d\n",item);
         break;
 Case3:
         displayQ();
         break;
 default: printf("End of operation");
          exit(0);
      }
   }
}
```