LinearQueue.c

Saved

```c
#include<stdio.h>
#define QUE_SIZE 3
int it,fr=0,rear=-1,q[10];
void insertrear()
{
  if(rear==QUE_SIZE-1)
  {
    printf("Queue Overflow\n");
    return;
  }
  rear+=1;
  q[rear]=it;
}
int deletefront()
{
  if(fr>rear)
  {
    fr=0;
    rear=-1;
    return -1;
  }
  return q[fr++];
}
void display()
{
  if(fr>rear)
  {
    printf("Queue is Empty\n");
    return;
  }
  printf("Contents of Queue:\n");
  for(int i=fr;i<=rear;i++)
    printf("%d\n",q[i]);
}
void main()
{
  int ch;
  for(;;)
  {
    printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit\n");
    printf("Enter Choice:");
    scanf("%d",&ch);
    switch(ch)
    {
      case 1:
        printf("\nEnter Item to be inserted:");
        scanf("%d",&it);
        insertrear();
        break;
      case 2:
        it=deletefront();
        if(it==-1)
          printf("\nEmpty Queue");
        else
          printf("Item Deleted=%d\n",it);
        break;
      case 3:
        display();
        break;
      default:exit(0);
    }
  }
}
```

```
Terminal

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:2

Empty Queue
1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:1

Enter Item to be inserted:11

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:1

Enter Item to be inserted:22

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:1

Enter Item to be inserted:33

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:1

Enter Item to be inserted:44
Queue Overflow

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:3
Contents of Queue:
11
22
33

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:2
Item Deleted=11

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:2
Item Deleted=22

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:2
Item Deleted=33

1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:2

Empty Queue
1.Insert
2.Delete
3.Display
4.Exit
Enter Choice:4

Process finished.
```

Lab Program - 3 : 14/10/20.

1BM19CS090
RAHIL

Write a program to simulate the working of queue of integers using an array. Provide the following operations.
a) Insert Rear b) Delete Front c) Display the contents.
The program should print the appropriate messages for a queue empty and queue full condition.

code

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#define QUE_SIZE 3
int item, front = 0, rear = -1, q[10];
void insertrear()
{
  if (rear == QUE_SIZE - 1)
  {
    printf("queue overflow\n");
    return;
  }
rear = rear + 1
q[rear] = item;
}
int deleteFront()
{
  if (front > rear)
  {
    front = 0;
    rear = -1;
    return -1;
  }
return q[front++];
}
```

```
void displayQ()
{
  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]);
  }
}
void main()
{
  int choice;
  for(;;)
  {
    printf("\n1. Insert Rear \n2. Delete Front \n3. Display \n4. Exit \n");
    printf("Enter Choice : ");
    scanf("%d", &choice);
    switch (choice)
    {
    case 1 : printf("Enter item to be inserted : \n");
             scanf("%d ", &item);
             insert rear();
             break;
    case 2 : item = delete front();
             if (item == -1)
             printf("Empty Queue\n");
             else
             printf("Item Deleted = %d \n", item);
             break;
    case 3: display(); break;
    default : exit(0);
    }
}
```