DS/circular-queue-using-array.c

```
//circular queue implementation using array.

#include <stdio.h>
#include <stdlib.h>

#define MAX_SIZE 10

int queue[MAX_SIZE];
int front = -1;
int rear = -1;

void enqueue(int x);
int dequeue();
void display();

int main(){
   int op,x;
   while(1){
      printf("1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.\nEnter Your choice: ");
      scanf("%d",&op);
      switch(op){
         case 1:
            if((front == 0 && rear == MAX_SIZE - 1) || (front == rear + 1)){
               printf("Overflow.\n");
            }else{
               printf("Enter the item: ");
               scanf("%d",&x);
               enqueue(x);
            }
            break;
         case 2:
            if(front == -1){
               printf("Underflow.\n");
            }else{
               x = dequeue();
               printf("Removed %d from queue.\n",x);
            }
            break;
         case 3:
            if(front == -1){
               printf("Queue is empty.\n");
            }else{
               printf("The elements of queue are: ");
               display();
            }
            break;
         case 4:
            exit(0);
         default:
            printf("Invalid Input.\n");
      }
   }
   return 0;
}

void enqueue(int x){
   if(front == -1){
      front = 0;
```

```
    }
    rear = (rear + 1) % MAX_SIZE;
    queue[rear] = x;
}

int dequeue(){
    int x = queue[front];
    if(front == rear){
        front = -1;
        rear = -1;
    }else{
        front = (front + 1) % MAX_SIZE;
    }
    return x;
}

void display(){
    int i;
    if(front <= rear){
        for(i = front; i <= rear; i++){
            printf("%d -> ",queue[i]);
        }
    }else{
        for(i = front; i < MAX_SIZE; i++){
            printf("%d -> ",queue[i]);
        }
        for(i = 0; i <= rear; i++){
            printf("%d -> ",queue[i]);
        }
    }
    printf("NULL\n");
}
```

OUTPUT

```
PS S:\WorkSpace\CollegeWork\DataStructure> gcc .\circular-queue-using-array.c
PS S:\WorkSpace\CollegeWork\DataStructure> ./a
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.
Enter Your choice: 1
Enter the item: 12
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.
Enter Your choice: 1
Enter the item: 13
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.
Enter Your choice: 1
Enter the item: 14
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.
Enter Your choice: 3
The elements of queue are: 12 -> 13 -> 14 -> NULL
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.
Enter Your choice: 2
Removed 12 from queue.
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit.
Enter Your choice: 4
PS S:\WorkSpace\CollegeWork\DataStructure>
```