queue-using-linkedlist.c

```
//write a c program to implement queue using linked list.

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

typedef struct queue{
    int data;
    struct queue* next;
} queue;

queue* head = NULL;

void enqueue(int data){
    queue* new_queue = (queue*)malloc(sizeof(queue));
    new_queue->data = data;
    new_queue->next = head;
     head = new_queue;
}

void dequeue(){
    if (!head) {
        printf("UnderFlow(No item found in this Queue).\n");
        return;
    }

    if (!head->next) {
        int data = head->data;
        head = NULL;
        printf("%d Removed.\n",data);
        return;
    }

    queue* current = head;
    while (current->next->next) {
        current = current->next;
    }

    int data = current->next->data;
    free(current->next);
    current->next = NULL;
    printf("%d Removed.\n",data);
    return;
}

void display(){
    if(head == NULL){
        printf("Underflow(Queue is Empty).\n");
        return;
    }
    queue* tmp = head;
    printf("Rear -> ");
    while(tmp  != NULL){
        printf("%d -> ",tmp->data);
        tmp = tmp->next;
    }
    printf("Front\n");
}

void main(){
    int data;
    int op;
    while(1){
        printf("1. enQueue, 2. deQueue, 3. Display, 4. Exit.\nEnter Your Choice: ");
        scanf("%d",&op);
        switch(op){
            case 1:
                printf("Enter the element: ");
                scanf("%d",&data);
                enqueue(data);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Oops..");
                exit(0);
                break;
            default:
                printf("Wrong Input.\n");
        }
    }
}
```

OUTPUT

```
PS S:\WorkSpace\CollegeWork\DataStructure> gcc .\queue-using-linked-list.c
PS S:\WorkSpace\CollegeWork\DataStructure> ./a
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 1 14 1 13 1 13 1 12 1 156 1 190 1 88 1 90
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 3
90 -> 88 -> 190 -> 156 -> 12 -> 13 -> 13 -> 14
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 2
Removed 14
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 2
Removed 13
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 3
90 -> 88 -> 190 -> 156 -> 12 -> 13
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 4
The len of this Queue is 6
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 5
The Rear of this Queue is 90
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 6
The Front of this Queue is 13
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 3
90 -> 88 -> 190 -> 156 -> 12 -> 13
1. enQueue, 2. deQueue, 3. Display, 4. Size, 5. Rear, 6. Front, 7. Exit.
Enter Your Choice: 7
Oops..
PS S:\WorkSpace\CollegeWork\DataStructure>
```