

# **CSE 2003: Lab Assignment #3**

Due on Thursday, March 1, 2017

*Shaik Naseera 2:00pm*

**Jacob John**

## Contents

<b>Problem 1</b>	<b>3</b>
<b>Problem 2</b>	<b>6</b>

## Problem 1

Write a program to implement Linear Queue using Array.

Listing 1: C program to implement Linear Queue using Array

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 10
int queue_arr[MAX];
5 int rear = -1;
int front = -1;
void insert(int item);
int del();
int peek();
10 void display();
int isFull();
int isEmpty();

int main()
15 {
    int choice,item;
    while(1)
    {
        printf("1.Enqueue\n");
        printf("2.Dequeue\n");
        printf("3.isFull\n");
        printf("4.isEmpty\n");
        printf("5.Display\n");
        printf("6.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
30         printf("Input the element for adding in queue : ");
            scanf("%d",&item);
            insert(item);
            break;

            case 2:
35         item=del();
            printf("Deleted element is %d\n",item);
            break;

            case 3:
40         if((isFull())==0)
            printf("Overflown: False\n");
            else
            printf("Overflown: True\n");
            break;
45         case 4:
            if((isEmpty())==0)
```

```

        printf("Not Empty\n");
50    else
        printf("Empty\n");
        break;

        case 5:
55    display();
        break;

        case 6:
        exit(1);
60    break;

        default:
        printf("Wrong choice\n")
        }/*End of switch*/
65    }/*End of while*/
    }/*End of main()*/

void insert(int item)
{
70    if(isFull())
        {
            printf("Queue Overflow\n");
            return;
        }
75    if(front== -1)
        front=0;
    rear=rear+1;
    queue_arr[rear] = item;
}/*End of insert()*/

80 int del()
{
    int item;
    if(isEmpty())
85    {
        printf("Queue Underflow\n"); exit(1);
    }
    item = queue_arr[front];
    front=front+1;
90    return item;
}/*End of del()*/

int isEmpty()
{
95    if(front == -1 || front == rear+1)
        return 1;
    else
        return 0;
}/*End of isEmpty()*/

100 int isFull()
```

```

{
    if(rear == MAX-1)
        return 1;
105    else
        return 0;
}/*End of isFull()*/

void display()
110 {
    int i;
    if(isEmpty())
    {
        printf("Queue is empty\n");
115        return;
    }
    printf("Queue is: \n\n");
    for(i=front; i<=rear; i++)
        printf("%d ",queue_arr[i]);
120    printf("\n\n");
}/*End of display*/

```

### Output:

```

3.isFull
4.isEmpty
5.Display
6.Quit
Enter your choice : 1
Input the element for adding in queue : 10
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Quit
Enter your choice : 2
Deleted element is 10
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Quit
Enter your choice : 3
Overflow: False
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Quit
Enter your choice : 4
Empty
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Quit
Enter your choice : 5
Queue is empty
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Quit
Enter your choice : 6
Jacobs-MacBook-Pro:Downloads jacobjohn$

```

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #define MAX 10
4 int queue_arr[MAX];
5 int rear = -1;
6 int front = -1;
7 void insert(int item);
8 int del();
9 int peek();
10 void display();
11 int isFull();
12 int isEmpty();
13
14 int main()
15 {
16     int choice,item;
17     while(1)
18     {
19         printf("1.Enqueue\n");
20         printf("2.Dequeue\n");
21         printf("3.isFull\n");
22         printf("4.isEmpty\n");
23         printf("5.Display\n");
24         printf("6.Quit\n");
25         printf("Enter your choice : ");
26         scanf("%d",&choice);
27         switch(choice)
28         {
29             case 1:
30                 printf("Input the element for adding in queue : ");
31                 scanf("%d",&item);
32                 insert(item);
33                 break;
34             case 2:
35                 item=del();
36                 printf("Deleted element is %d\n",item);
37                 break;
38             case 3:
39                 if((isFull())==0)
40                     printf("Overflow: False\n");
41                 else
42                     printf("Overflow: True\n");
43                 break;
44             case 4:
45                 if((isEmpty())==0)
46                     printf("Queue is empty\n");
47                 else
48                     printf("Queue is not empty\n");
49                 break;
50             case 5:
51                 display();
52                 break;
53             case 6:
54                 break;
55         }
56     }
57 }

```

## Problem 2

Write a program to implement Circular Queue using Array.

Listing 2: C program to implement Circular Queue using Array

```
#include<stdio.h>
#include<stdlib.h>
#define max_size 10
int cqueue[max_size], front=-1, rear=-1;
5 void insert();
void del();
void display();
int isFull();
int isEmpty();
10 int main()
{
    int choice;
    do{
        printf("\n\n-----CIRCULAR QUEUE OPERATIONS-----\n");
        printf("1.Enqueue\n");
        printf("2.Dequeue\n");
        printf("3.isFull\n");
        printf("4.isEmpty\n");
        20 printf("5.Display\n");
        printf("6.Exit\n");
        printf("-----");
        printf("\nEnter your choice:\t");
        scanf("%d",&choice);
        25 switch(choice)
        {
            case 1:
                insert();
                break;
            case 2:
                del();
                break;
            case 5:
                35 display();
                break;
            case 6:
                40 exit(0);
                break;
            case 3:
                if((isFull())==0)
                45 printf("Overflown: False\n");
                else
                printf("Overflown: True\n");
                break;
```

```
50         case 4:
            if ((isEmpty())==0)
                printf("Not Empty\n");
            else
                printf("Empty\n");
55         break;

        default:
            printf("\nInvalid choice:\n");
            break;
60     }
    } while (choice!=6);

}

void insert() //Inserting an element in to the queue
65 {
    int item;
    if (front==(rear+1)%max_size)
    {
        printf("\nQueue Overflow:");
70     }
    else
    {
        printf("Enter the element to be inserted:\t");
        scanf("%d",&item);
75         rear=(rear+1)%max_size;
        cqueue[rear]=item;

        if (front==-1)
        {
80             front=0;
            rear=0;
        }

    }
85 }

//end of insert()

void del() //deleting an element from the queue
{
90     int item;
    if (front==-1)
    {
        printf("\nQueue Underflow:");
    }
95     else
    {
        item=cqueue[front];
        printf("\nThe deleted element: %d\t",item);
        if (front==rear)
100        {
            front=-1;
        }
    }
}
```

```

        rear=-1;
    }
    else
    {
        front=(front+1)%max_size;
    }

}

110 }//end of del()

void display() //To display the queue elements
{
    int i;
    if(front==-1)
    {
        printf("\nQueue is Empty:");
    }
    else
    {
        printf("\nThe queue elements are:\n" );
        if (front<rear)
        {
            for (i=front;i<=rear;i++)
            {
                printf("%d\t",cqueue[i]);
            }
        }
        else
        {
            for (i=0;i<=rear;i++)
            {
                printf("%d\t",cqueue[i]);
            }
            for (i=front;i<max_size;i++)
            {
                printf("%d\t",cqueue[i]);
            }
        }
    }

}

140 }//end of display()

145 int isEmpty()
{
    if (front==-1) return 1;
    else
        return 0;
150 }/*End of isEmpty()*/

int isFull()
{
    if((front==0 && rear==max_size-1)|| (front==rear+1))

```



```

155         return 1;
        else
            return 0;
    }/*End of isFull()*/

```

**Output:**

The screenshot shows a terminal window with the following output:

```

-----CIRCULAR QUEUE OPERATIONS-----
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Exit
-----
Enter your choice: 1
Enter the element to be inserted: 10

-----CIRCULAR QUEUE OPERATIONS-----
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Exit
-----
Enter your choice: 1
Enter the element to be inserted: 30

-----CIRCULAR QUEUE OPERATIONS-----
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Exit
-----
Enter your choice: 1
Enter the element to be inserted: 40

-----CIRCULAR QUEUE OPERATIONS-----
1.Enqueue
2.Dequeue
3.isFull
4.isEmpty
5.Display
6.Exit
-----

```

The background code is the C implementation of a circular queue:

```

circular_queue.c
single_linked_list.c
linear_queue.c
+
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define max_size 10
4 int cqueue[max_size],front=-1,rear=-1;
5 void insert();
6 void del();
7 void display();
8 int isFull();
9 int isEmpty();
10 int main()
11 {
12     int choice;
13     do{
14
15         printf("\n\n-----CIRCULAR QUEUE OPERATIONS-----\n\n");
16         printf("1.Enqueue\n");
17         printf("2.Dequeue\n");
18         printf("3.isFull\n");
19         printf("4.isEmpty\n");
20         printf("5.Display\n");
21         printf("6.Exit\n");
22         printf("-----");
23         printf("\nEnter your choice:\t");
24         scanf("%d",&choice);
25         switch(choice)
26         {
27             case 1:
28                 insert();
29                 break;
30
31             case 2:
32                 del();
33                 break;
34
35             case 5:
36                 display();
37                 break;
38
39             case 6:
40                 exit(0);
41                 break;
42
43             case 3:
44                 if((isFull()==0))
45                     printf("Overflown: False\n");
46                 else
47                     printf("Overflown: True\n");
48         }
49     }while(choice!=6);
50     return 0;
51 }

```

Line: 17:39 | C | Tab Size: 4 | main