

## **Double Circular Queue**

**จัดทำโดย**

นายปิยภูมิ มีคำบาง 6530200282

นายศุภชัย แก้วละมูล 6530200509

นายภาณุพงศ์ ทองเจ็ด 6530200339

นายกษิตศ อยู่คง 6530200576

นายสิทธิเดช เลิศล้ำนา 6530200533

**เสนอ**

ผศ.ดร. จีรวรรณ เจริญสุข

คณะวิทยาศาสตร์ ศรีราชา

มหาวิทยาลัย เกษตรศาสตร์ วิทยาเขตศรีราชา

ไฟล์เราใช้ในการทำโปรแกรม Double Circular Queue มีทั้งหมดแค่ 1 ไฟล์เท่านั้นคือ  
“ProjectDataStruct.c”

สามารถเข้าถึงได้เพื่อ GitHub นี้เพื่ออ่าน Source Code

[https://github.com/Piyaphum/DoubleCircularQueue\\_2](https://github.com/Piyaphum/DoubleCircularQueue_2)

## ตัวอย่างหน้าจอเมื่อใช้โปรแกรม

### 1. หน้าจอ Main Menu

```
*****MENU*****  
1. Enqueue_front  
2. Enqueue_rear  
3. Dequeue_front  
4. Dequeue_rear  
5. Exit  
Enter your choice: |
```

## 2. หน้าจอ EnQueue\_Front

```
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 1

Enter item value (char or int): ijk
Queue elements: i

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: j i

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: |
```

### 3. หน้าจอ EnQueue\_Rear

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 2

Enter item value (char or int): abc
Queue elements: k j i a

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i a b

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i a b c

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
```

#### 4. หน้าจอ DeQueue\_Front

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 3
k has been dequeued
Queue elements: j i a b c
```

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: ijk
j has been dequeued
Queue elements: i a b c
```

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: i has been dequeued
Queue elements: a b c
```

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: a has been dequeued
```

```
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: a has been dequeued
Queue elements: b c
```

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: b has been dequeued
Queue elements: c
```

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: c has been dequeued
Queue is empty.
```

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
UNDERFLOW
```

```
-----
Process exited after 110.2 seconds with return value 3221225477
Press any key to continue . . . |
```

5. หน้าจอ DeQueue\_Rear

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i a b c

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 4
c has been dequeued
Queue elements: k j i a b

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: |
```



6. หน้าจอเมื่อออกจากโปรแกรมอย่างถูกต้อง

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i a b c

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 4
c has been dequeued
Queue elements: k j i a b

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 5

-----
Process exited after 39.7 seconds with return value 0
Press any key to continue . . . |
```

## 7. หน้าจอเมื่อใส่ค่าผิด

```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 00000

Wrong selection!!! Try again!!!

*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: |
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node
5  {
6      struct node* prev;
7      struct node* next;
8      char data; // Changed from int to char
9  };
10
11 struct node* head = NULL;
12 struct node* rear = NULL;
13
14 void enqueue_front();
15 void enqueue_rear();
16 void dequeue_front();
17 void dequeue_rear();
18 void display();
19
20 int main() {
21     int choice;
22     while(1) {
23         printf("\n\n*****MENU*****\n");
24         printf("1. Enqueue_front\n2. Enqueue_rear\n3. Dequeue_front\n4. Dequeue_rear\n5. Exit\n");
25         printf("Enter your choice: ");
26         scanf("%d", &choice);
27
28         switch(choice) {
29             case 1:
30                 enqueue_front();
31                 display();
32                 break;
33             case 2:
34                 enqueue_rear();
35                 display();
36                 break;
37             case 3:
38                 dequeue_front();
39                 display();
40                 break;
41             case 4:
42                 dequeue_rear();
43                 display();
44                 break;
45             case 5:
46                 exit(0);
47             default:
48                 printf("\nWrong selection!!! Try again!!!");
49         }
50     }
51 }
```

```

1 void enqueue_front() {
2     struct node* newnode;
3     char item;
4     newnode = (struct node*)malloc(sizeof(struct node));
5     if(newnode == NULL) {
6         printf("\nOVERFLOW");
7     }
8     else {
9         printf("\nEnter item value (char or int): ");
10        if (scanf(" %c", &item) == 1) { // Changed to read a character
11            newnode->data = item;
12        }
13        if (head == NULL) {
14            head = rear = newnode;
15            newnode->next = newnode;
16            newnode->prev = newnode;
17        }
18        else {
19            newnode->next = head;
20            newnode->prev = rear;
21            rear->next = newnode;
22            head->prev = newnode;
23            head = newnode;
24        }
25    }
26 }
27
28 void enqueue_rear() {
29     struct node* newnode;
30     char item;
31     newnode = (struct node*)malloc(sizeof(struct node));
32     if(newnode == NULL) {
33         printf("\nOVERFLOW");
34     }
35     else {
36         printf("\nEnter item value (char or int): ");
37         if (scanf(" %c", &item) == 1) { // Changed to read a character
38             newnode->data = item;
39         }
40         if (head == NULL) {
41             head = rear = newnode;
42             newnode->next = newnode;
43             newnode->prev = newnode;
44         }
45         else {
46             newnode->next = head;
47             newnode->prev = rear;
48             rear->next = newnode;
49             head->prev = newnode;
50             rear = newnode;
51         }
52     }
53 }

```



```
1  void deQueue_front() {
2      struct node* temp = head;
3      if (head == NULL) {
4          printf("\nUNDERFLOW");
5      }
6      else {
7          if (head == rear) {
8              head = rear = NULL;
9          }
10         else {
11             head = head->next;
12             head->prev = rear;
13             rear->next = head;
14         }
15     }
16     printf("%c has been dequeued\n", temp->data);
17     free(temp);
18 }
19
20 void deQueue_rear() {
21     struct node* temp = rear;
22     if (head == NULL) {
23         printf("\nUNDERFLOW");
24     }
25     else {
26         if (head == rear) {
27             head = rear = NULL;
28         }
29         else {
30             rear = rear->prev;
31             rear->next = head;
32             head->prev = rear;
33         }
34     }
35     printf("%c has been dequeued\n", temp->data);
36     free(temp);
37 }
```



```
1 void display() {  
2     if (head == NULL) {  
3         printf("Queue is empty.\n");  
4         return;  
5     }  
6  
7     struct node* current = head;  
8     printf("Queue elements: ");  
9     do {  
10        printf("%c ", current->data);  
11        current = current->next;  
12    } while (current != head);  
13    printf("\n");  
14 }
```