# Double Circular Queue

**จัดทำโดย**

นายปิยภูมิ มืดคำบง 6530200282

นายศุภชัย แก้วละมุล 6530200509

นายภาณุพงศ์ ทองเชิด 6530200339

นายกษิดิศ อยู่คง 6530200576

นายสิทธิเดช เลิศลัคนา 6530200533

**เสนอ**

ผศ.ดร. จิรวรรณ เจริญสุข

คณะวิทยาศาสตร์ ศรีราชา

มหาวิทยาลัย เกษตรศาสตร์ วิทยาเขตศรีราชา

ไฟล์เราใช้ในการทำโปรแกรม Double Circular Queue มีทั้งหมดแค่ 1 ไฟล์เท่านั้นคือ

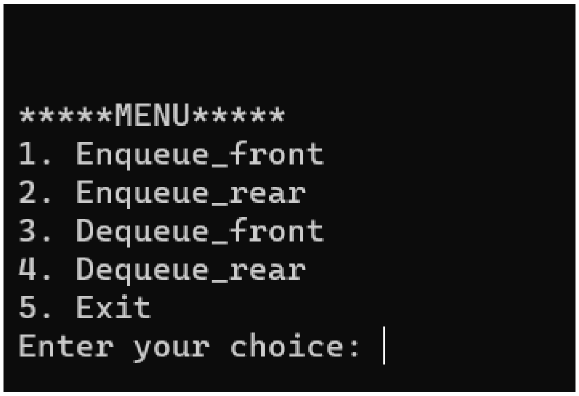
“ProjectDataStruct.c”

สามารถเข้าลิงค์ได้เพื่อ GitHub นี้เพื่ออ่าน Source Code

https://github.com/Piyaphum/DoubleCircularQueue_2

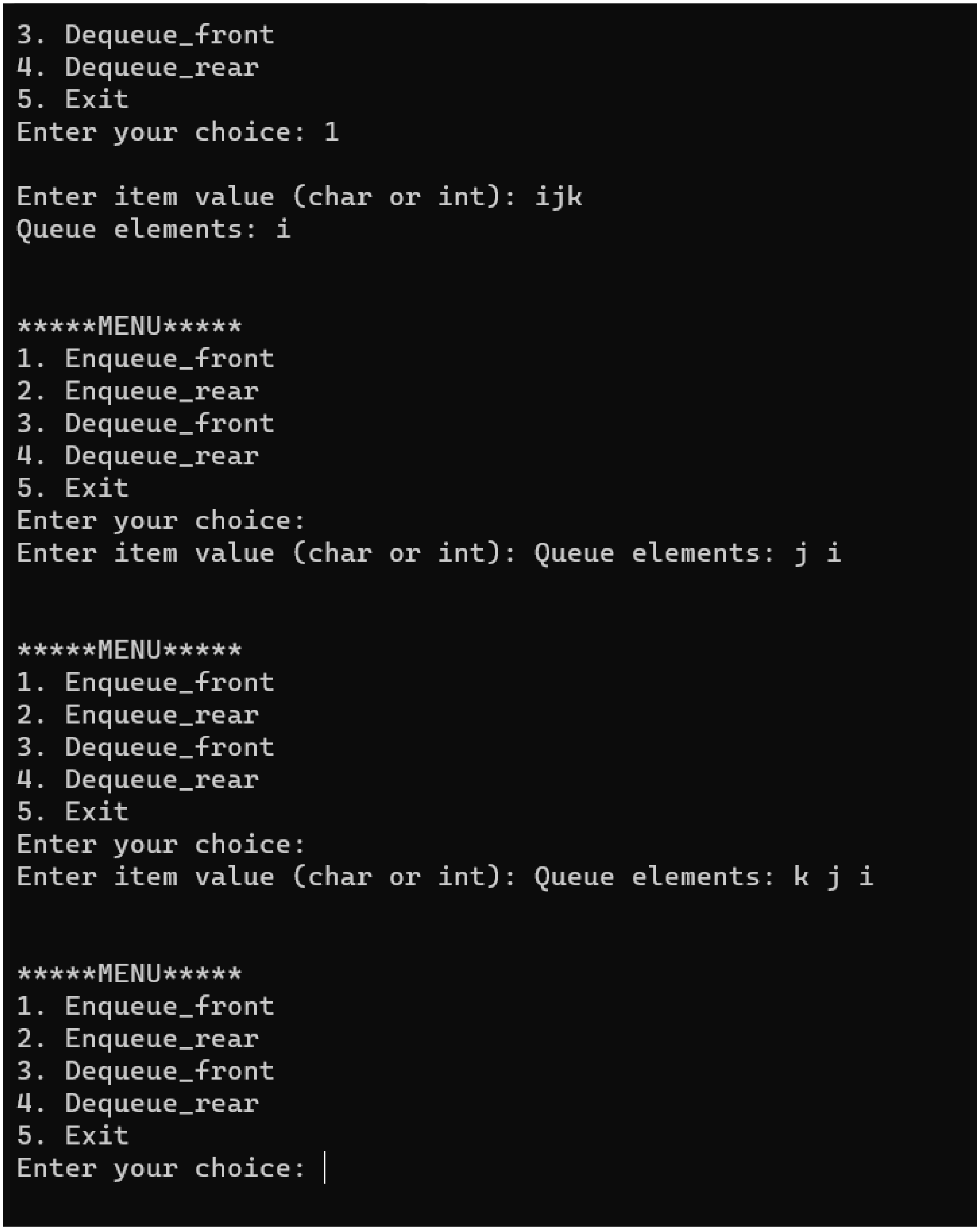
# ตัวอย่างหน้าจอเมื่อใช้โปรแกรม

1. หน้าจอ Main Menu



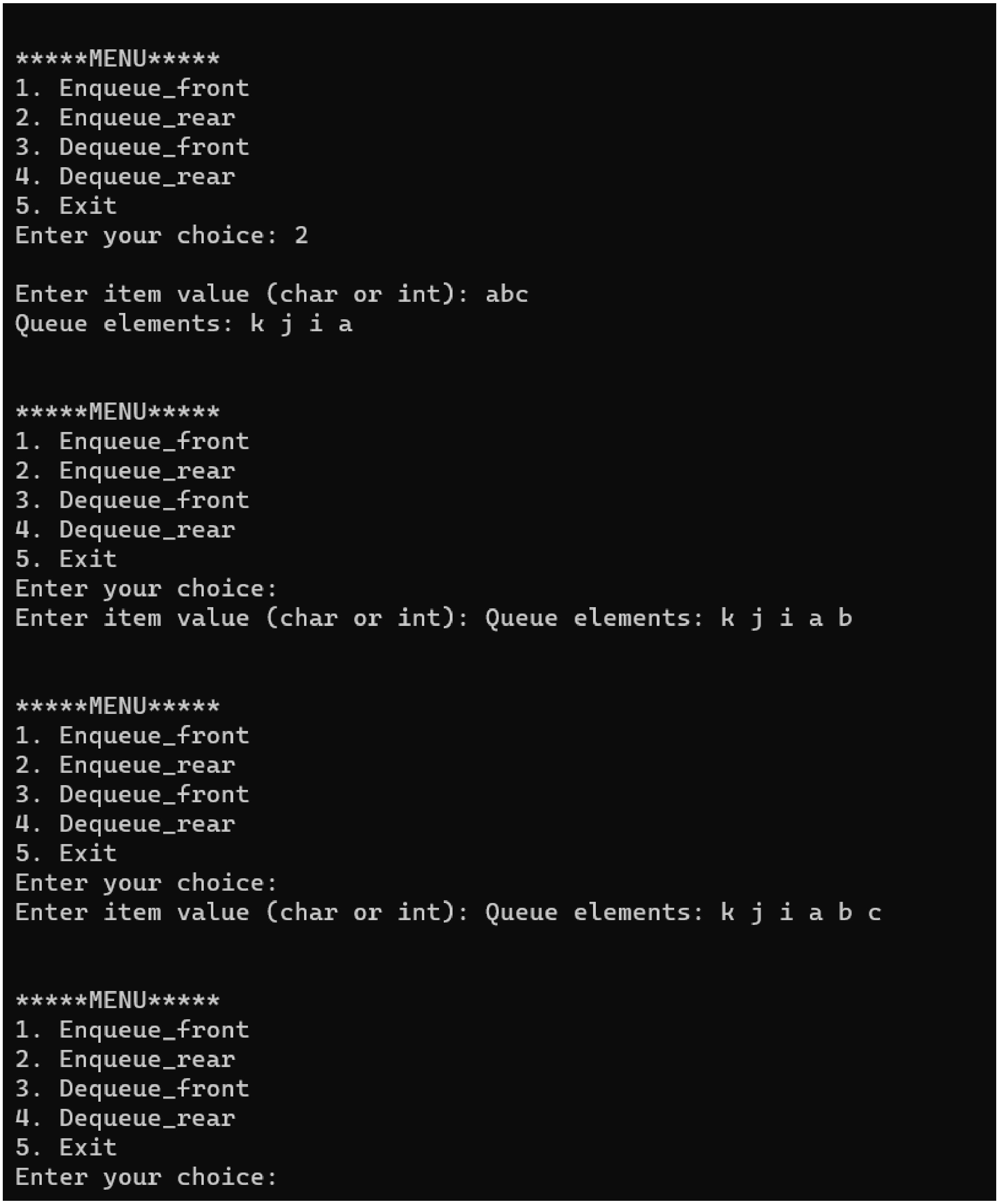
```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 
```

2. หน้าจอ EnQueue_Front



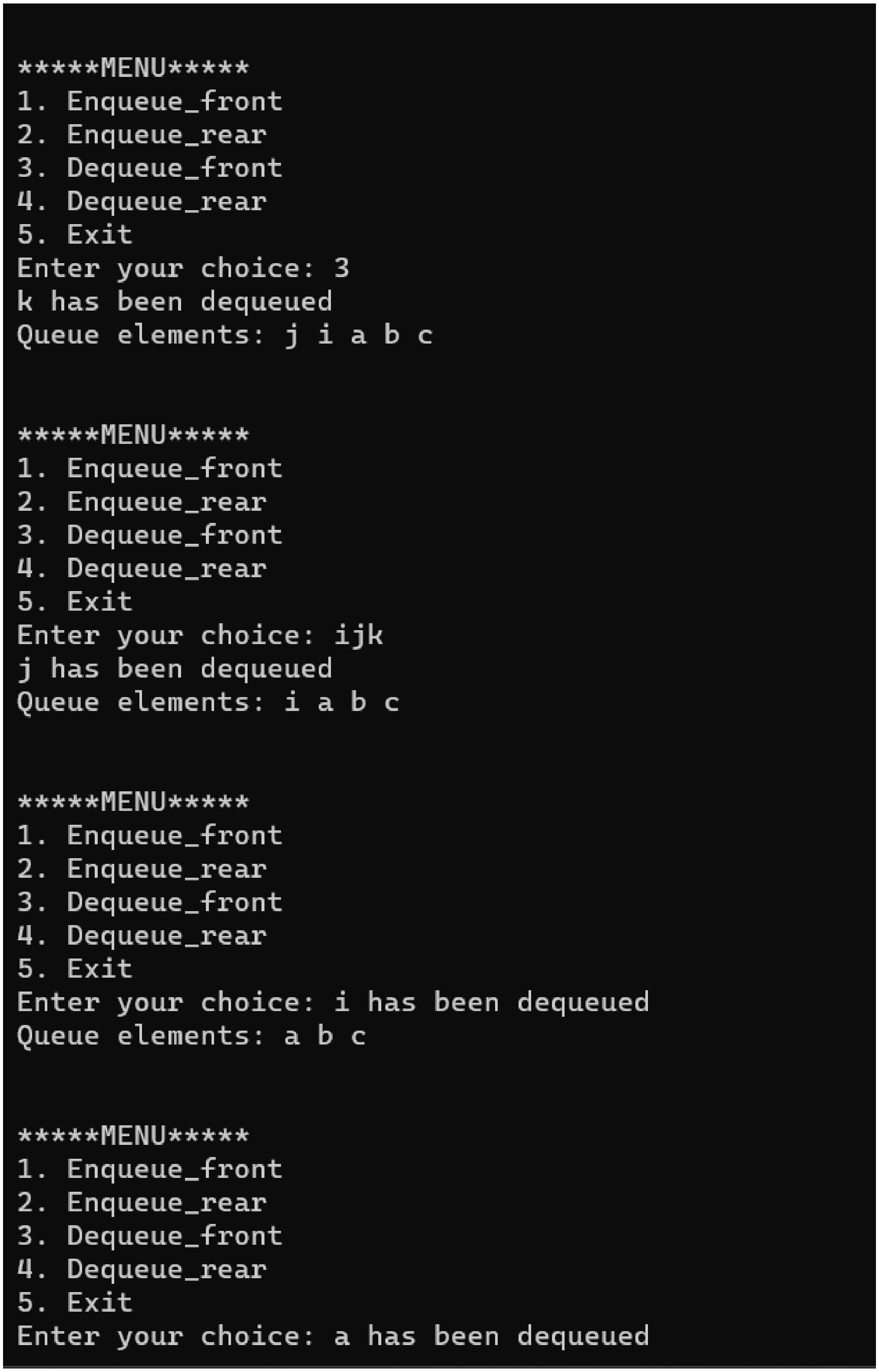
```
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 1

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

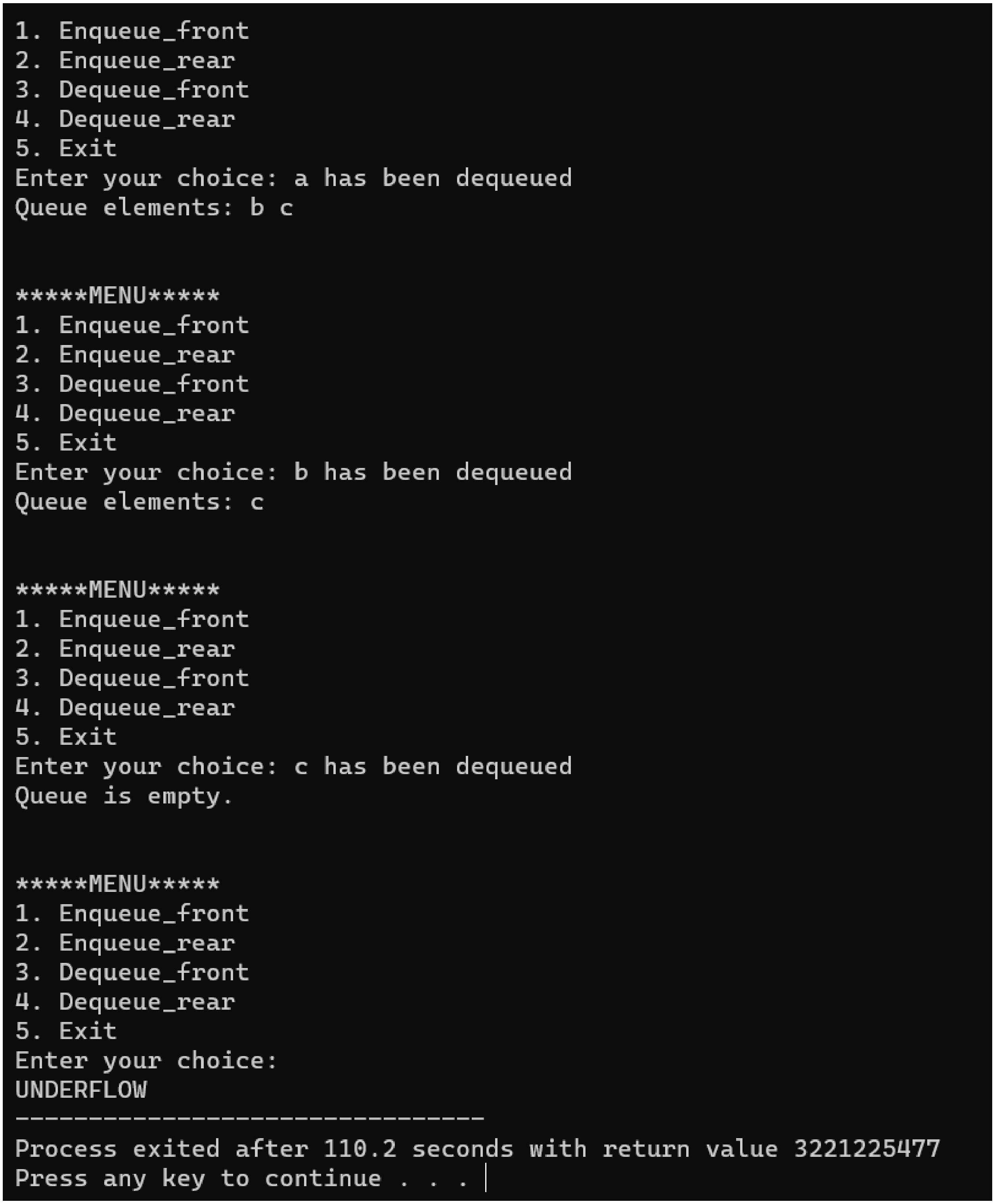


```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 2

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

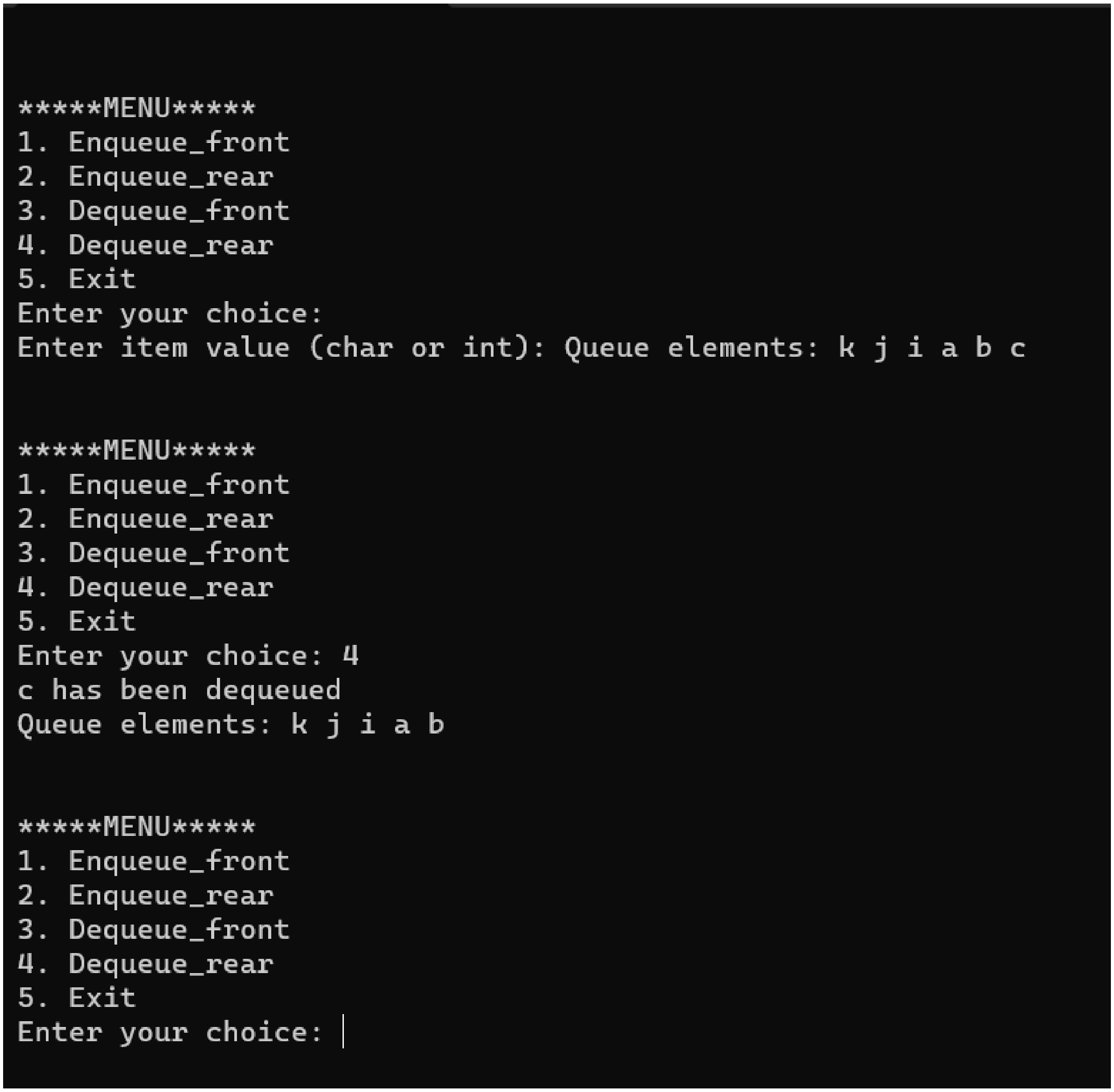


```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: 3
k has been dequeued
Queue elements: j i a b c


*****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
```



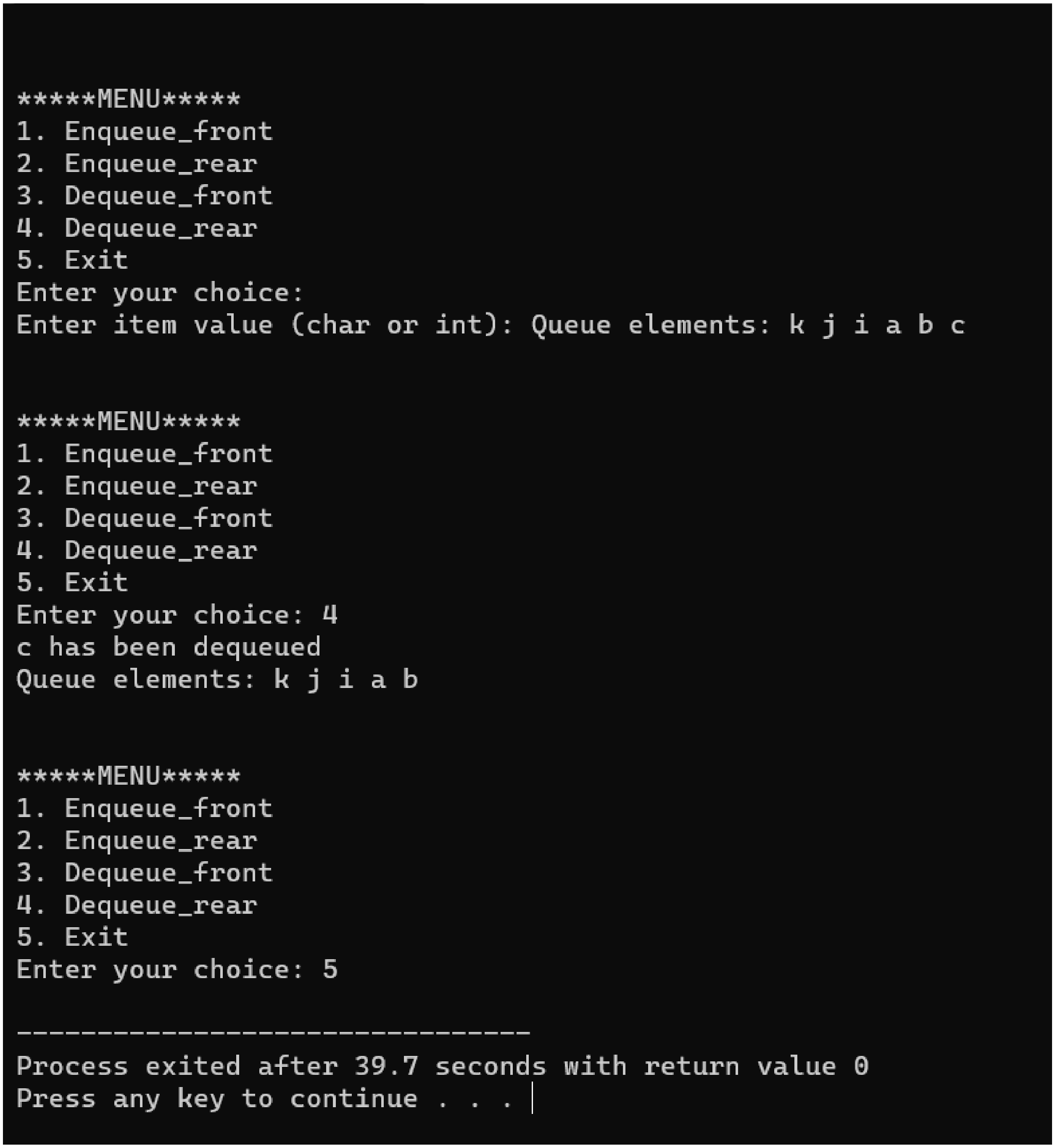
```
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice: a has been dequeued
Queue elements: b c


*****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



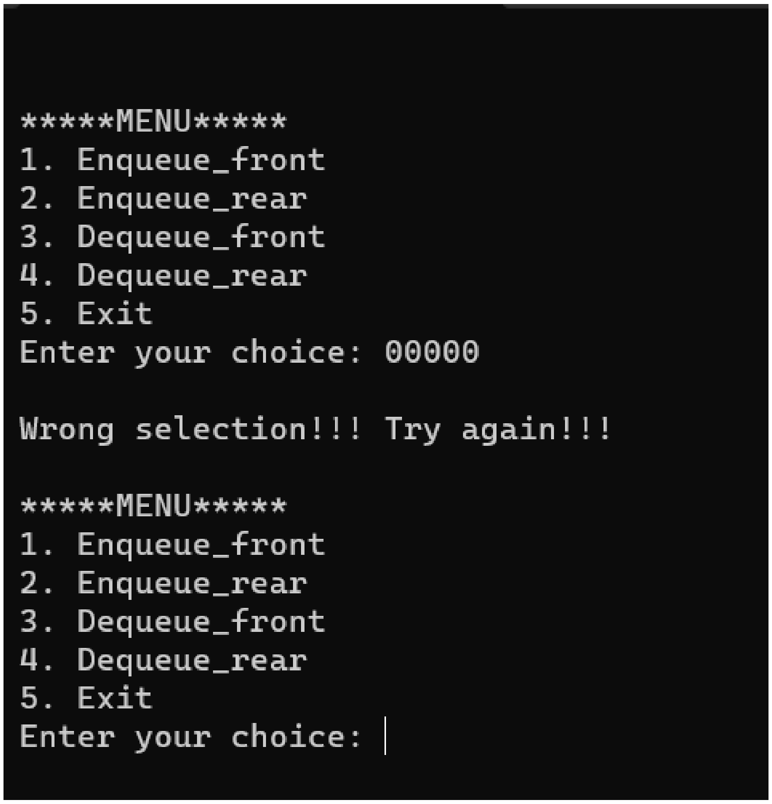
```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i a b c


*****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

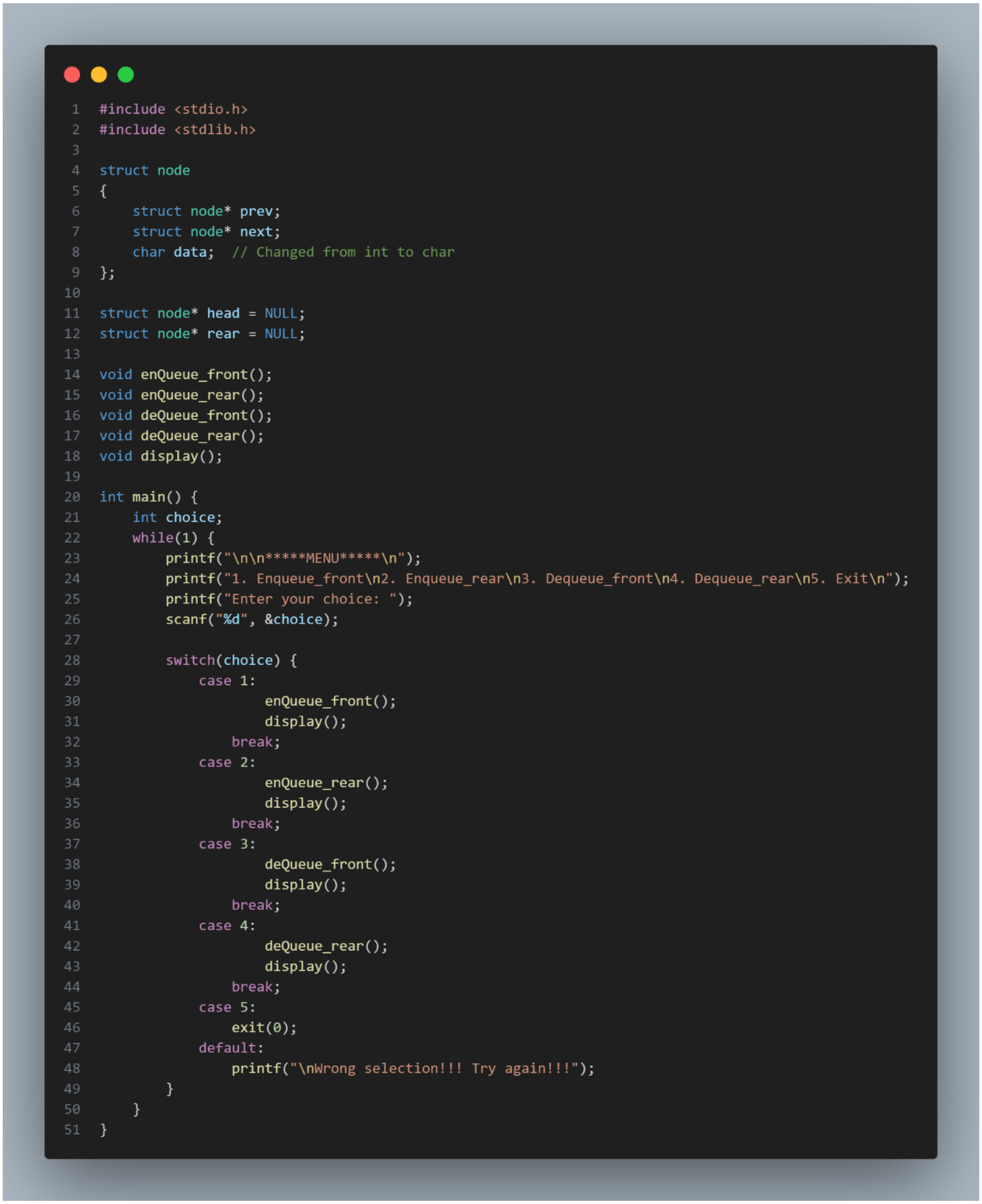


```
*****MENU*****
1. Enqueue_front
2. Enqueue_rear
3. Dequeue_front
4. Dequeue_rear
5. Exit
Enter your choice:
Enter item value (char or int): Queue elements: k j i a b c


*****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 . . .
```

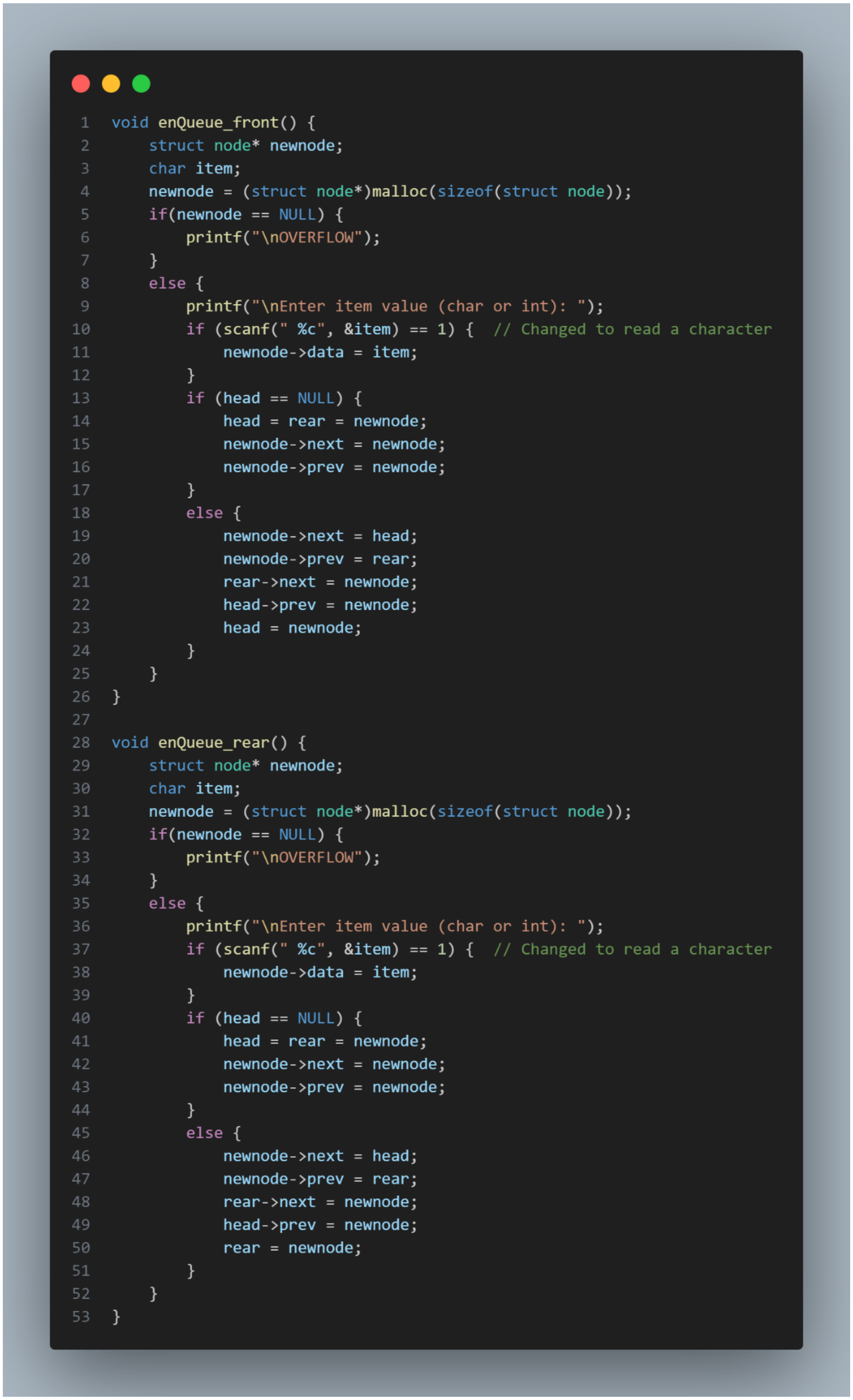
7. หน้าจอเมื่อใส่ค่าผิด



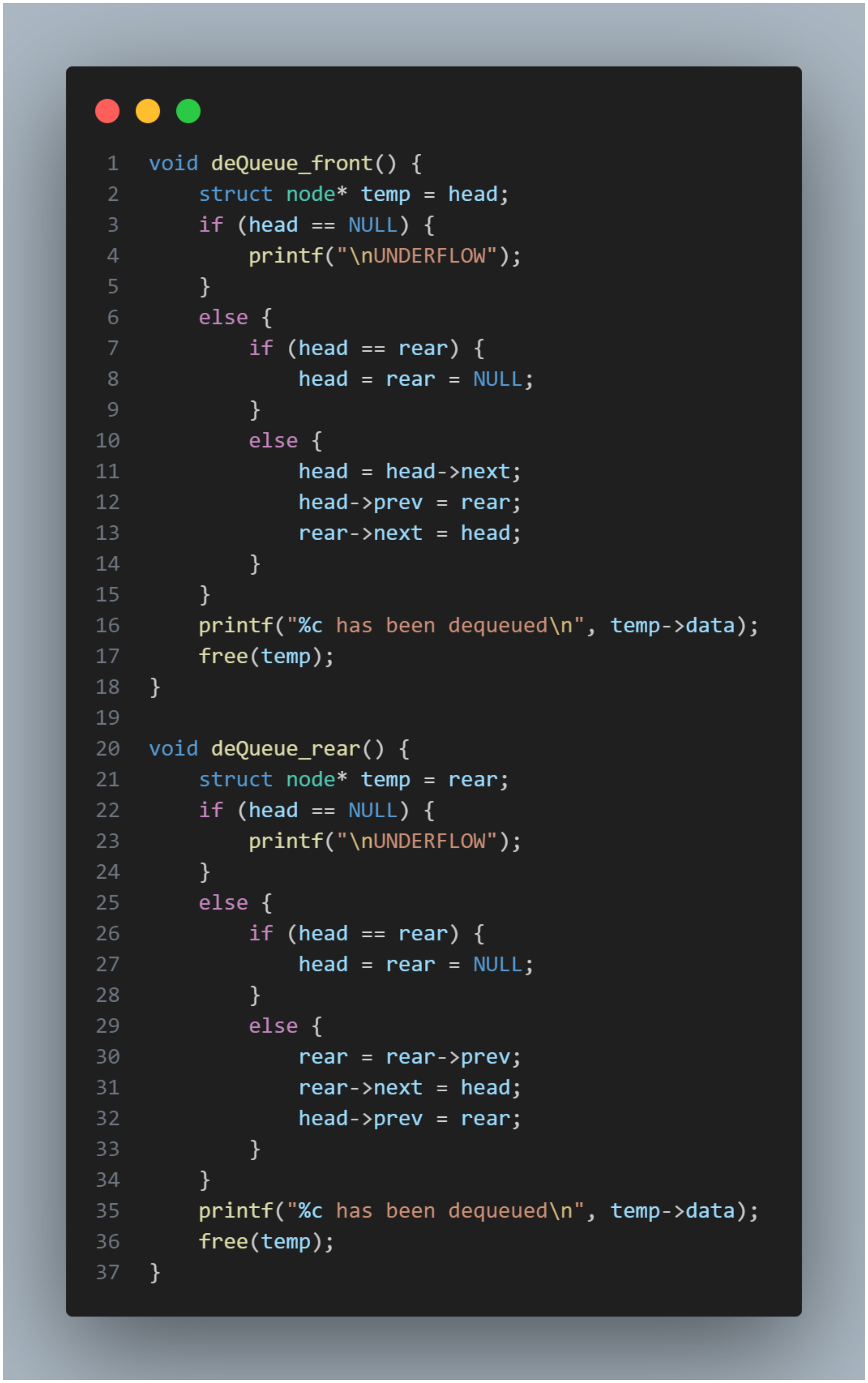
```


*****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: 
```



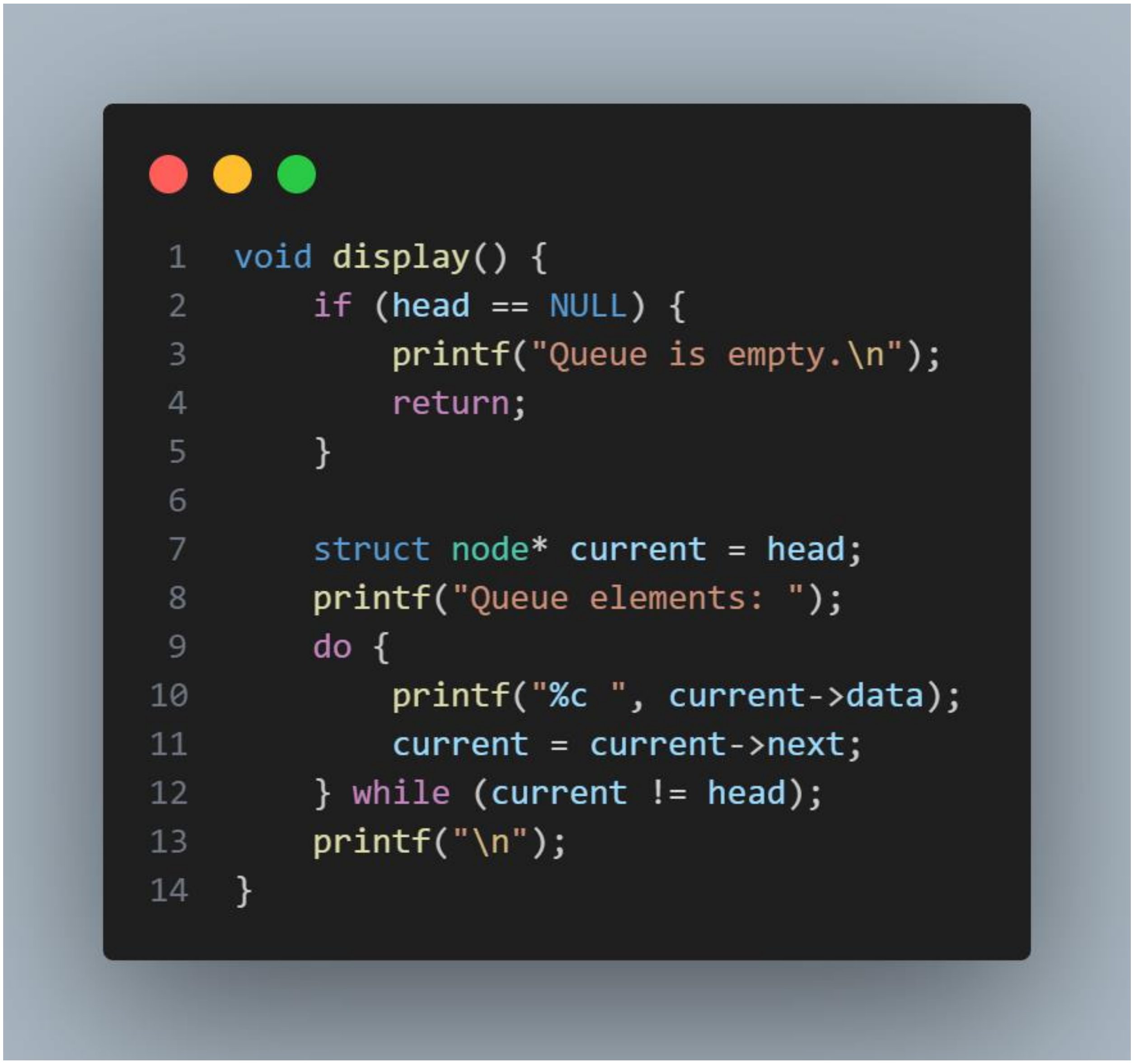
```c
#include <stdio.h>
#include <stdlib.h>

struct node
{
    struct node* prev;
    struct node* next;
    char data;  // Changed from int to char
};

struct node* head = NULL;
struct node* rear = NULL;

void enQueue_front();
void enQueue_rear();
void deQueue_front();
void deQueue_rear();
void display();

int main() {
    int choice;
    while(1) {
        printf("\n\n*****MENU*****\n");
        printf("1. Enqueue_front\n2. Enqueue_rear\n3. Dequeue_front\n4. Dequeue_rear\n5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1:
                    enQueue_front();
                    display();
                break;
            case 2:
                    enQueue_rear();
                    display();
                break;
            case 3:
                    deQueue_front();
                    display();
                break;
            case 4:
                    deQueue_rear();
                    display();
                break;
            case 5:
                exit(0);
            default:
                printf("\nWrong selection!!! Try again!!!");
        }
    }
}
```



```c
void enQueue_front() {
    struct node* newnode;
    char item;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL) {
        printf("\nOVERFLOW");
    }
    else {
        printf("\nEnter item value (char or int): ");
        if (scanf(" %c", &item) == 1) {  // Changed to read a character
            newnode->data = item;
        }
        if (head == NULL) {
            head = rear = newnode;
            newnode->next = newnode;
            newnode->prev = newnode;
        }
        else {
            newnode->next = head;
            newnode->prev = rear;
            rear->next = newnode;
            head->prev = newnode;
            head = newnode;
        }
    }
}

void enQueue_rear() {
    struct node* newnode;
    char item;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL) {
        printf("\nOVERFLOW");
    }
    else {
        printf("\nEnter item value (char or int): ");
        if (scanf(" %c", &item) == 1) {  // Changed to read a character
            newnode->data = item;
        }
        if (head == NULL) {
            head = rear = newnode;
            newnode->next = newnode;
            newnode->prev = newnode;
        }
        else {
            newnode->next = head;
            newnode->prev = rear;
            rear->next = newnode;
            head->prev = newnode;
            rear = newnode;
        }
    }
}
```



```c
void deQueue_front() {
    struct node* temp = head;
    if (head == NULL) {
        printf("\nUNDERFLOW");
    }
    else {
        if (head == rear) {
            head = rear = NULL;
        }
        else {
            head = head->next;
            head->prev = rear;
            rear->next = head;
        }
    }
    printf("%c has been dequeued\n", temp->data);
    free(temp);
}

void deQueue_rear() {
    struct node* temp = rear;
    if (head == NULL) {
        printf("\nUNDERFLOW");
    }
    else {
        if (head == rear) {
            head = rear = NULL;
        }
        else {
            rear = rear->prev;
            rear->next = head;
            head->prev = rear;
        }
    }
    printf("%c has been dequeued\n", temp->data);
    free(temp);
}
```



```c
void display() {
    if (head == NULL) {
        printf("Queue is empty.\n");
        return;
    }

    struct node* current = head;
    printf("Queue elements: ");
    do {
        printf("%c ", current->data);
        current = current->next;
    } while (current != head);
    printf("\n");
}
```