자료구조론 CC343_2207

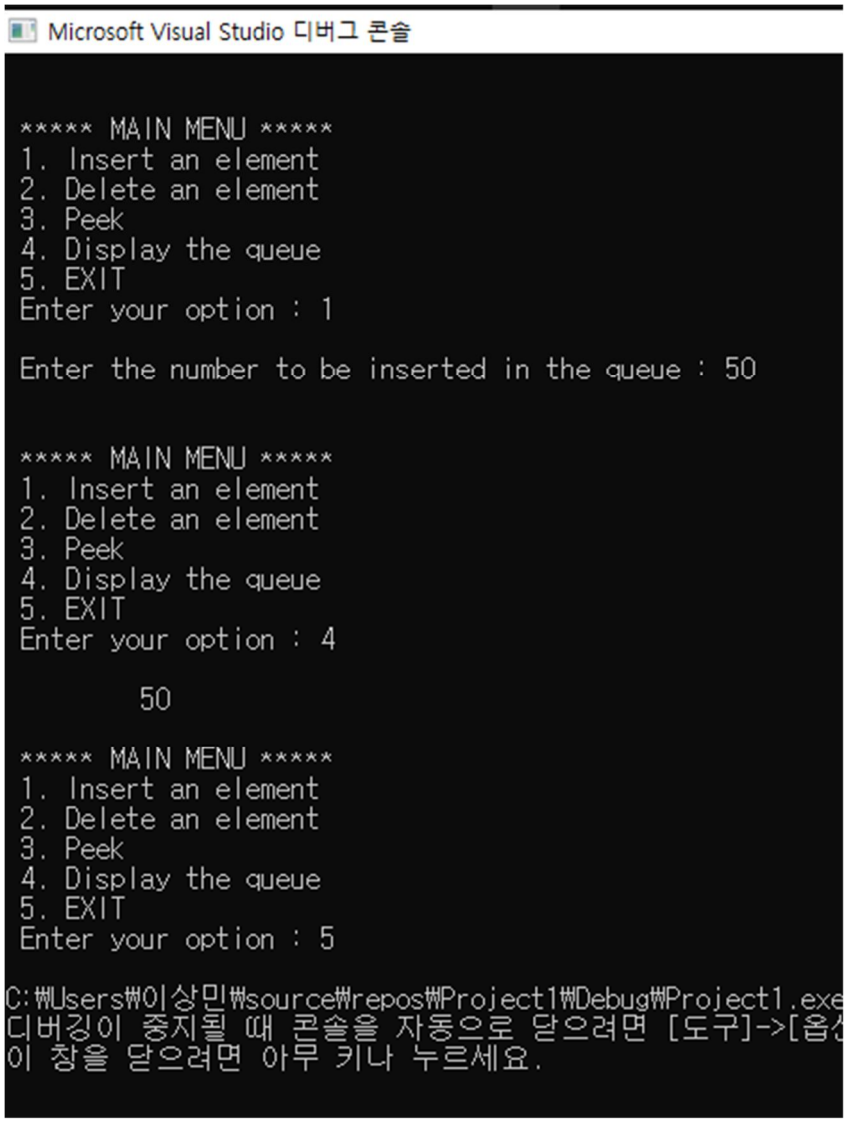
Programming assignment 6

경기대학교 컴퓨터공학부

201511837 이상민

Programming Example

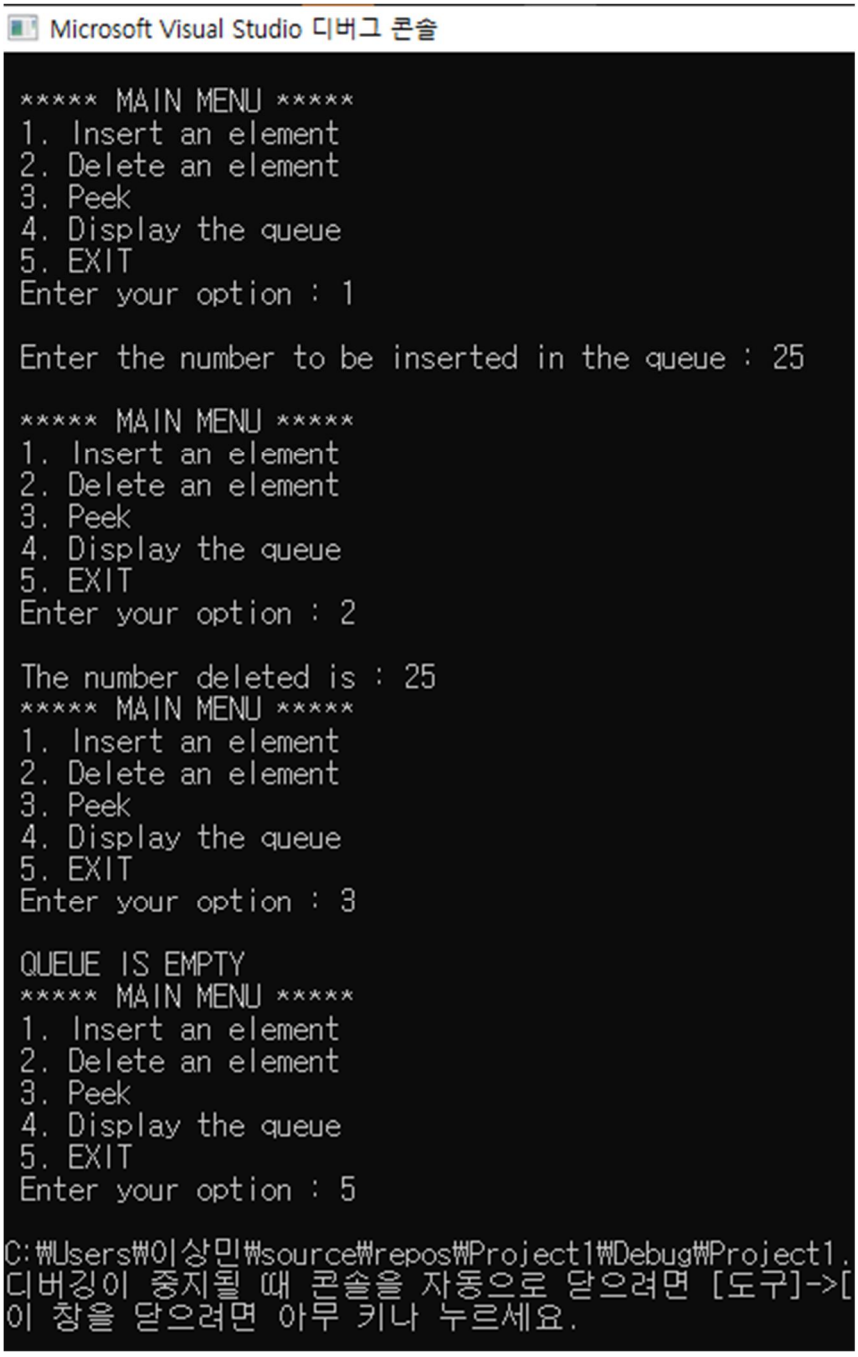
1번



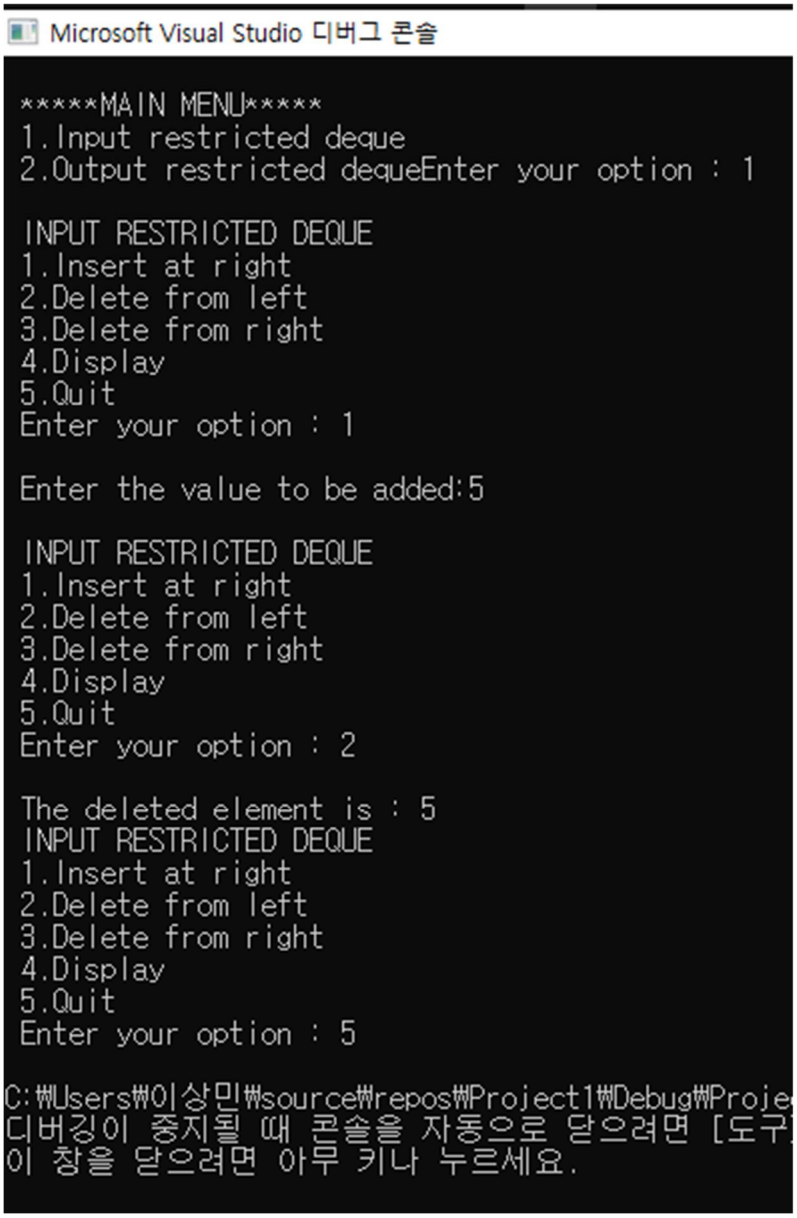
2번


Microsoft Visual Studio 디버그 콘솔
*****MAIN MENU*****
1. INSERT
2. DELETE
3. PEEK
4. DISPLAY
5. EXIT
Enter your option : 3
QUEUE IS EMPTY
*****MAIN MENU*****
1. INSERT
2. DELETE
3. PEEK
4. DISPLAY
5. EXIT
Enter your option : 5
C:₩Users₩이상민₩source₩repos₩Project1₩D
디버깅이 중지될 때 콘솔을 자동으로 닫으
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.

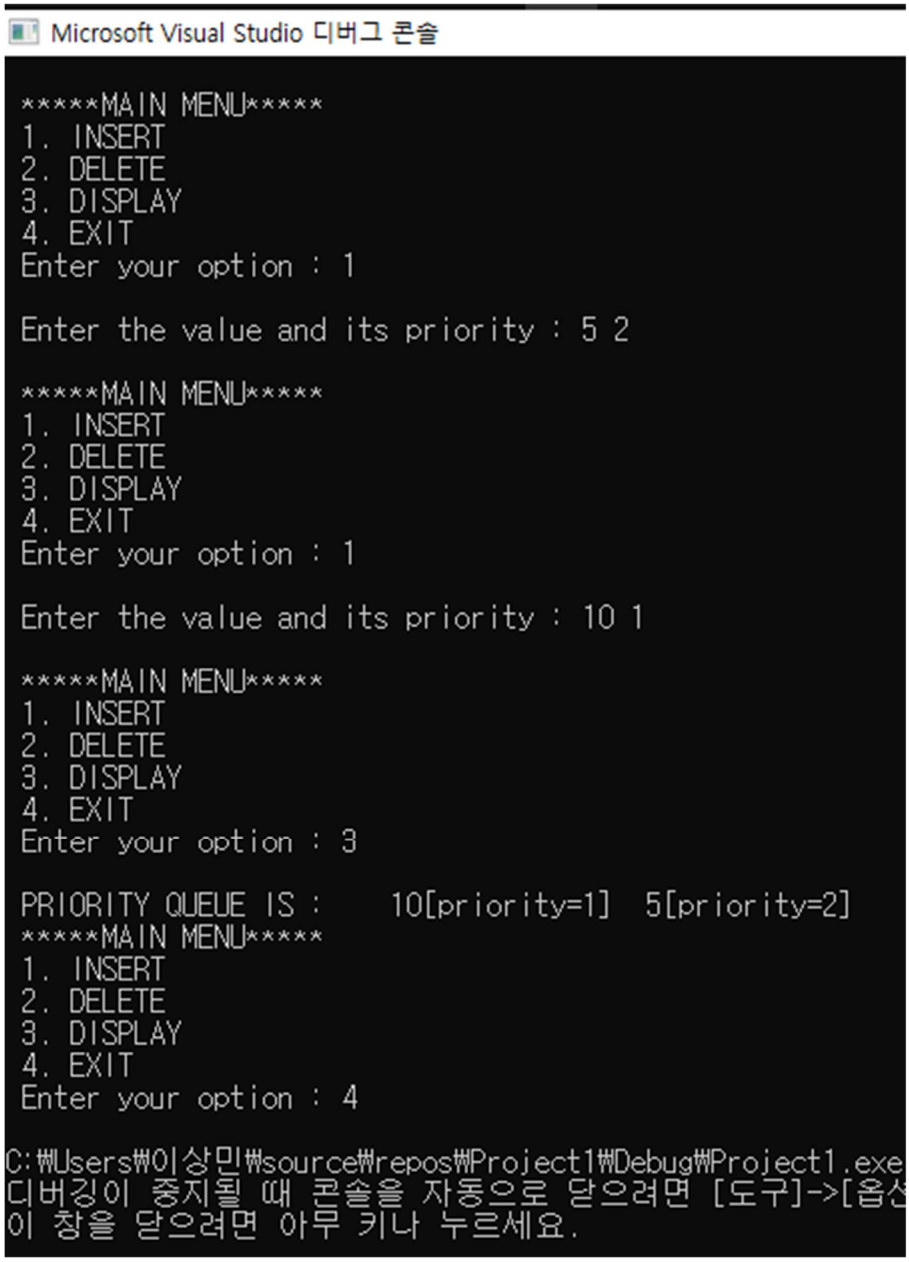
3번


Microsoft Visual Studio 디버그 콘솔
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 1
Enter the number to be inserted in the queue : 25
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 2
The number deleted is : 25
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 3
QUEUE IS EMPTY
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Peek
4. Display the queue
5. EXIT
Enter your option : 5
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Project1.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[
이 창을 닫으려면 아무 키나 누르세요.

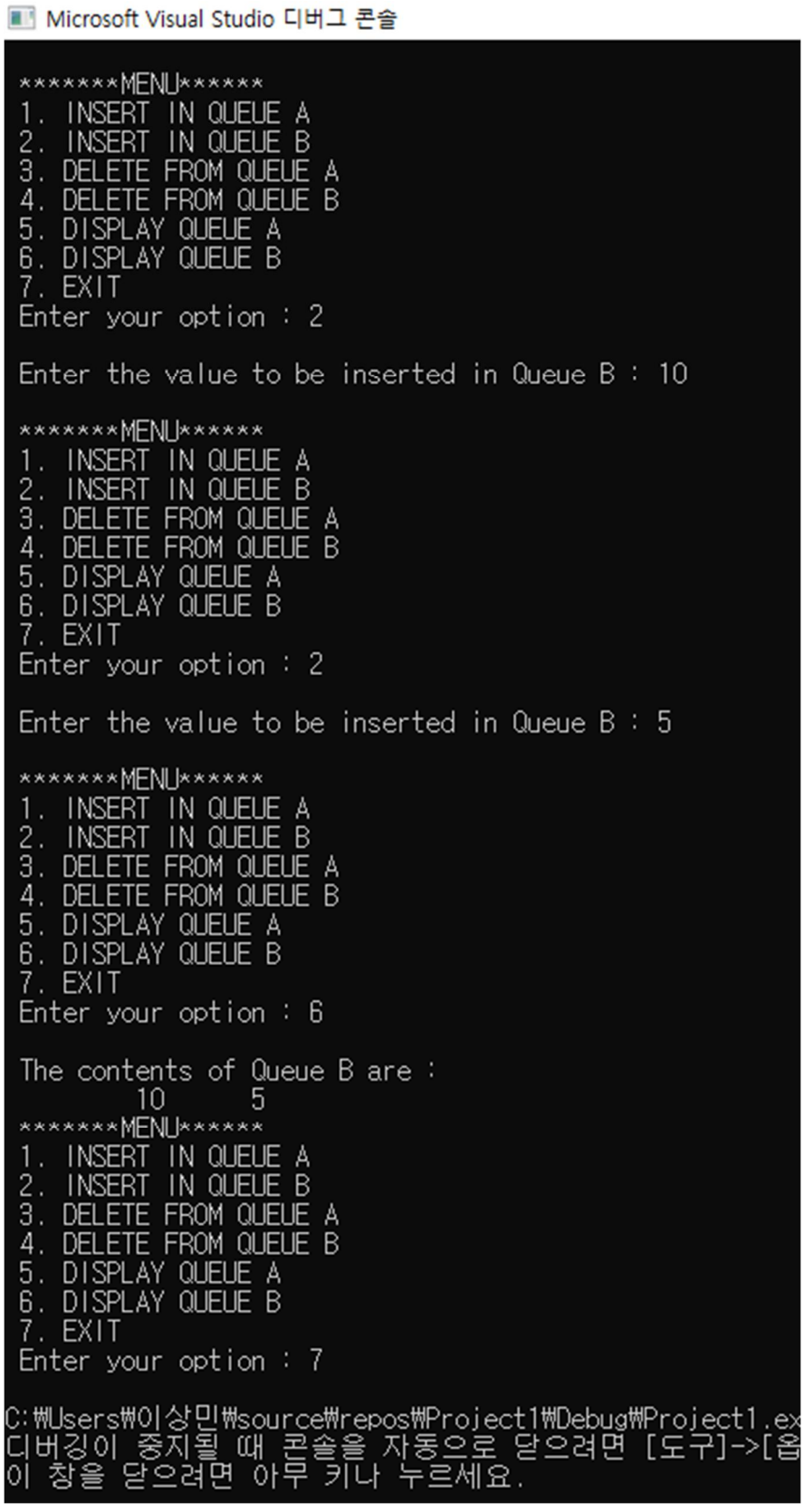
4번


Microsoft Visual Studio 디버그 콘솔
*****MAIN MENU*****
1.Input restricted deque
2.Output restricted dequeEnter your option : 1
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 1
Enter the value to be added:5
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 2
The deleted element is : 5
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 5
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Proje
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구
이 창을 닫으려면 아무 키나 누르세요.

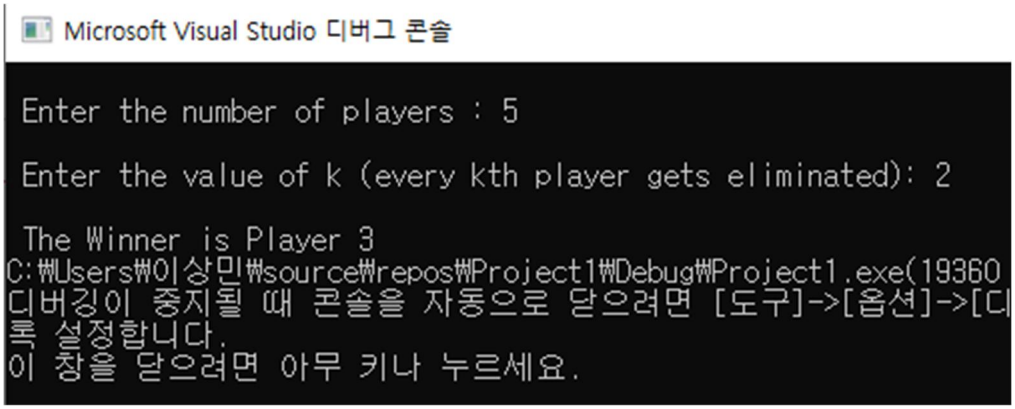
5번


Microsoft Visual Studio 디버그 콘솔
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 1
Enter the value and its priority : 5 2
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 1
Enter the value and its priority : 10 1
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 3
PRIORITY QUEUE IS : 10[priority=1] 5[priority=2]
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 4
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Project1.exe
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션
이 창을 닫으려면 아무 키나 누르세요.

6번

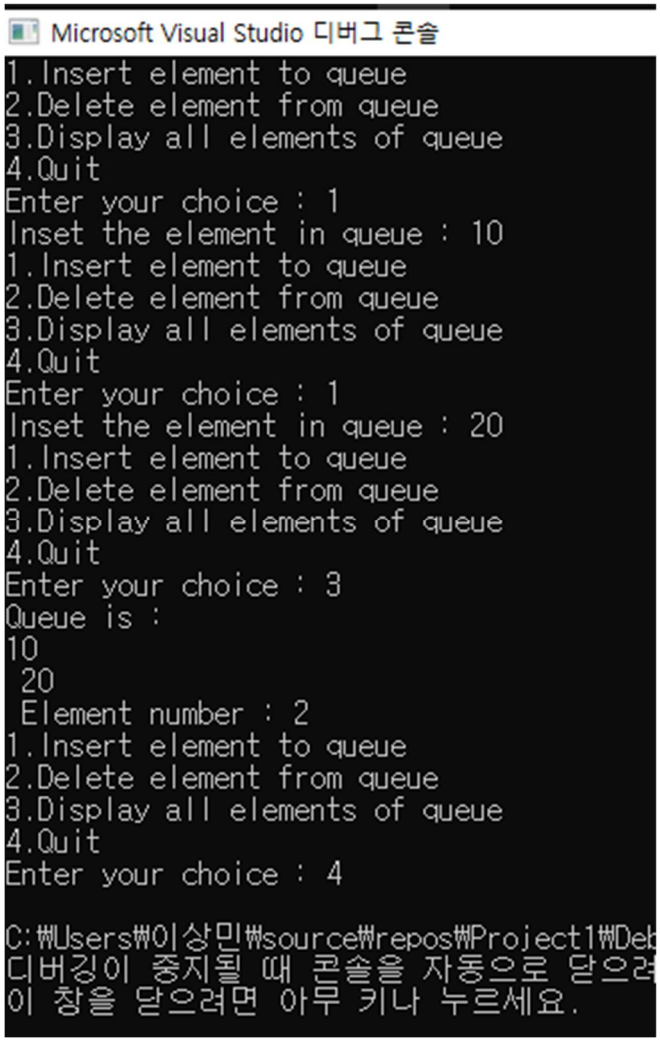

Microsoft Visual Studio 디버그 콘솔
*******MENU******
1. INSERT IN QUEUE A
2. INSERT IN QUEUE B
3. DELETE FROM QUEUE A
4. DELETE FROM QUEUE B
5. DISPLAY QUEUE A
6. DISPLAY QUEUE B
7. EXIT
Enter your option : 2
Enter the value to be inserted in Queue B : 10
*******MENU******
1. INSERT IN QUEUE A
2. INSERT IN QUEUE B
3. DELETE FROM QUEUE A
4. DELETE FROM QUEUE B
5. DISPLAY QUEUE A
6. DISPLAY QUEUE B
7. EXIT
Enter your option : 2
Enter the value to be inserted in Queue B : 5
*******MENU******
1. INSERT IN QUEUE A
2. INSERT IN QUEUE B
3. DELETE FROM QUEUE A
4. DELETE FROM QUEUE B
5. DISPLAY QUEUE A
6. DISPLAY QUEUE B
7. EXIT
Enter your option : 6
The contents of Queue B are :
10 5
*******MENU******
1. INSERT IN QUEUE A
2. INSERT IN QUEUE B
3. DELETE FROM QUEUE A
4. DELETE FROM QUEUE B
5. DISPLAY QUEUE A
6. DISPLAY QUEUE B
7. EXIT
Enter your option : 7
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Project1.ex
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵
이 창을 닫으려면 아무 키나 누르세요.

7번


Microsoft Visual Studio 디버그 콘솔
Enter the number of players : 5
Enter the value of k (every kth player gets eliminated): 2
The Winner is Player 3
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Project1.exe(19360
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.

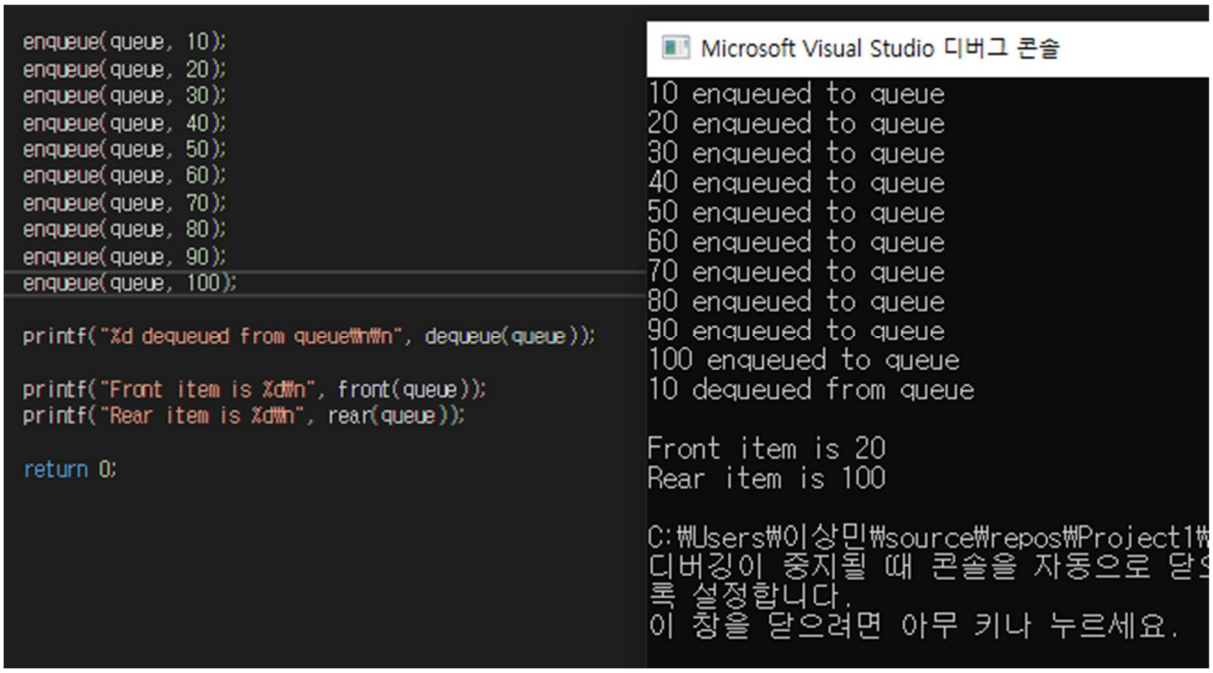
Programming exercises

1. Write a program to calculate the number of items in a queue.
대기열의 항목 수를 계산하는 프로그램을 작성한다.



2. Write a program to create a linear queue of 10 values .

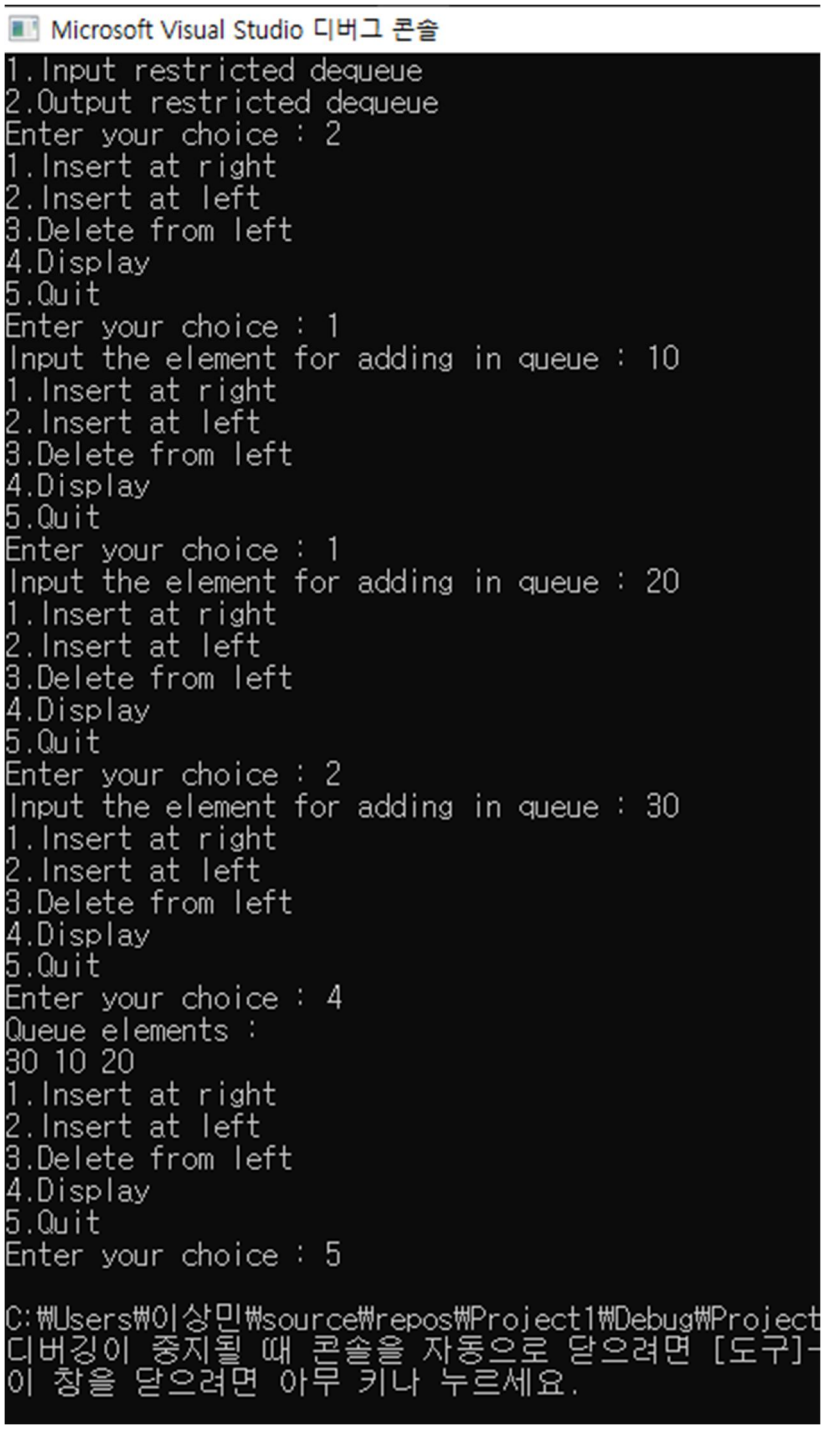
프로그램을 작성하여 10개의 값의 선형 대기열을 만든다.



```
enqueue(queue, 10);
enqueue(queue, 20);
enqueue(queue, 30);
enqueue(queue, 40);
enqueue(queue, 50);
enqueue(queue, 60);
enqueue(queue, 70);
enqueue(queue, 80);
enqueue(queue, 90);
enqueue(queue, 100);

printf("%d dequeued from queue₩n₩n", dequeue(queue));

printf("Front item is %d₩n", front(queue));
printf("Rear item is %d₩n", rear(queue));

return 0;
```

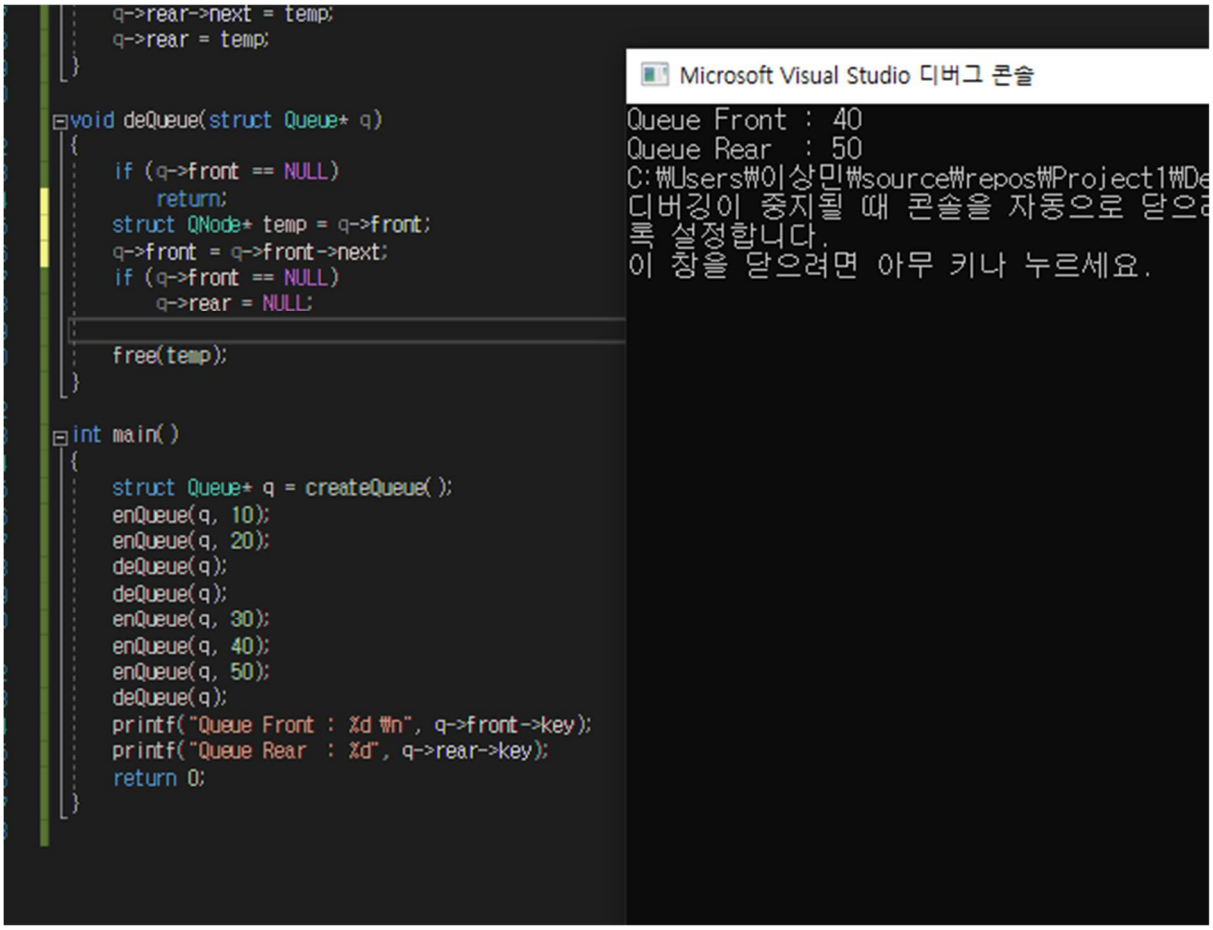
```
Microsoft Visual Studio 디버그 콘솔
10 enqueued to queue
20 enqueued to queue
30 enqueued to queue
40 enqueued to queue
50 enqueued to queue
60 enqueued to queue
70 enqueued to queue
80 enqueued to queue
90 enqueued to queue
100 enqueued to queue
10 dequeued from queue

Front item is 20
Rear item is 100

C:₩Users₩이상민₩source₩repos₩Project1₩
디버깅이 중지될 때 콘솔을 자동으로 닫
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

3. Write a program to create a queue using arrays which permits insertion at both the ends.
프로그램을 작성하여 양쪽 끝에 삽입이 가능한 배열로 큐를 만든다.

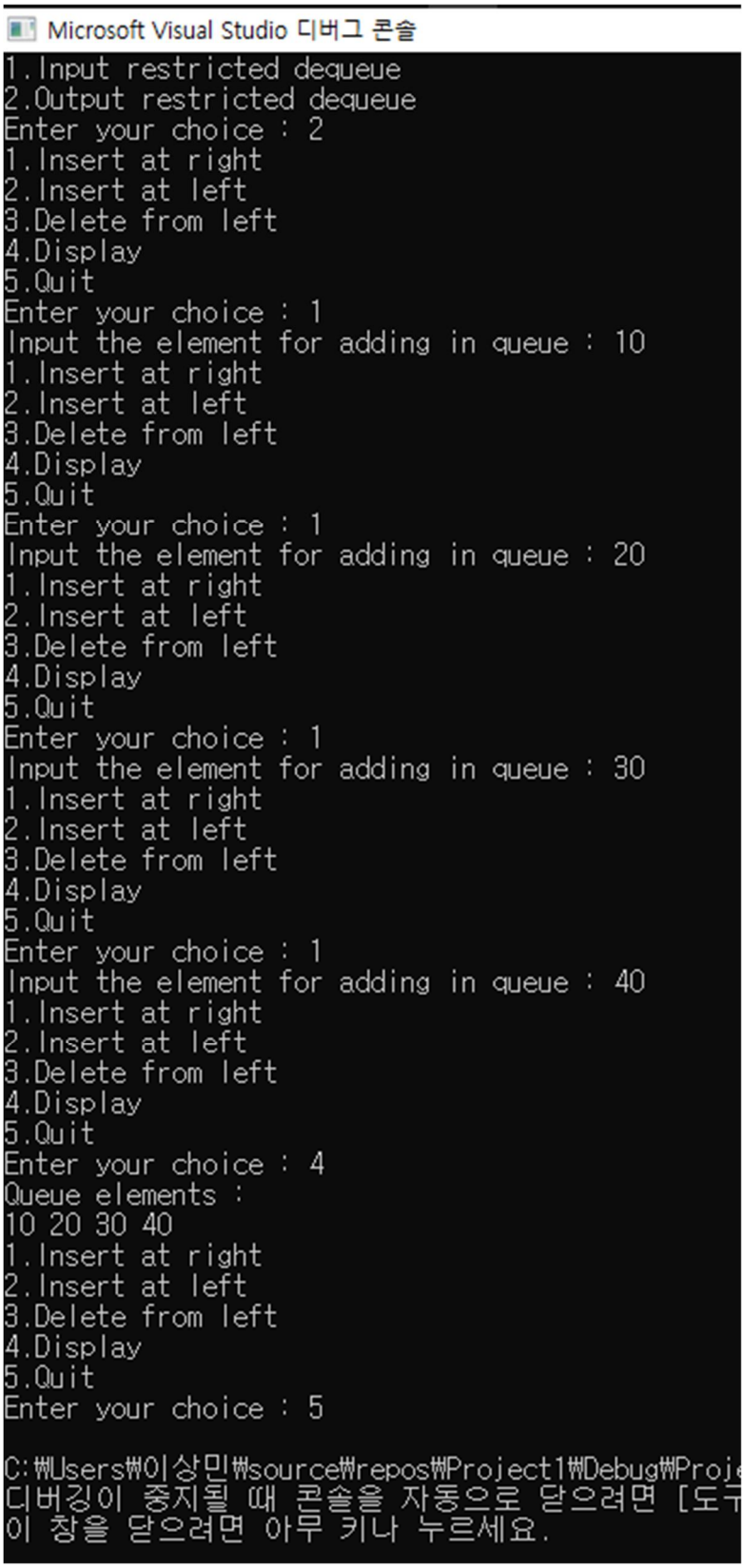


4. Write a program to implement a dequeue with the help of a linked list.

링크된 목록의 도움을 받아 dequeue를 구현하는 프로그램을 작성한다.



5. Write a program to create a queue which permits insertion at any vacant location at the rear end.
프로그램을 작성하여 뒤쪽 끝의 빈 위치에 삽입할 수 있는 대기열을 만든다.



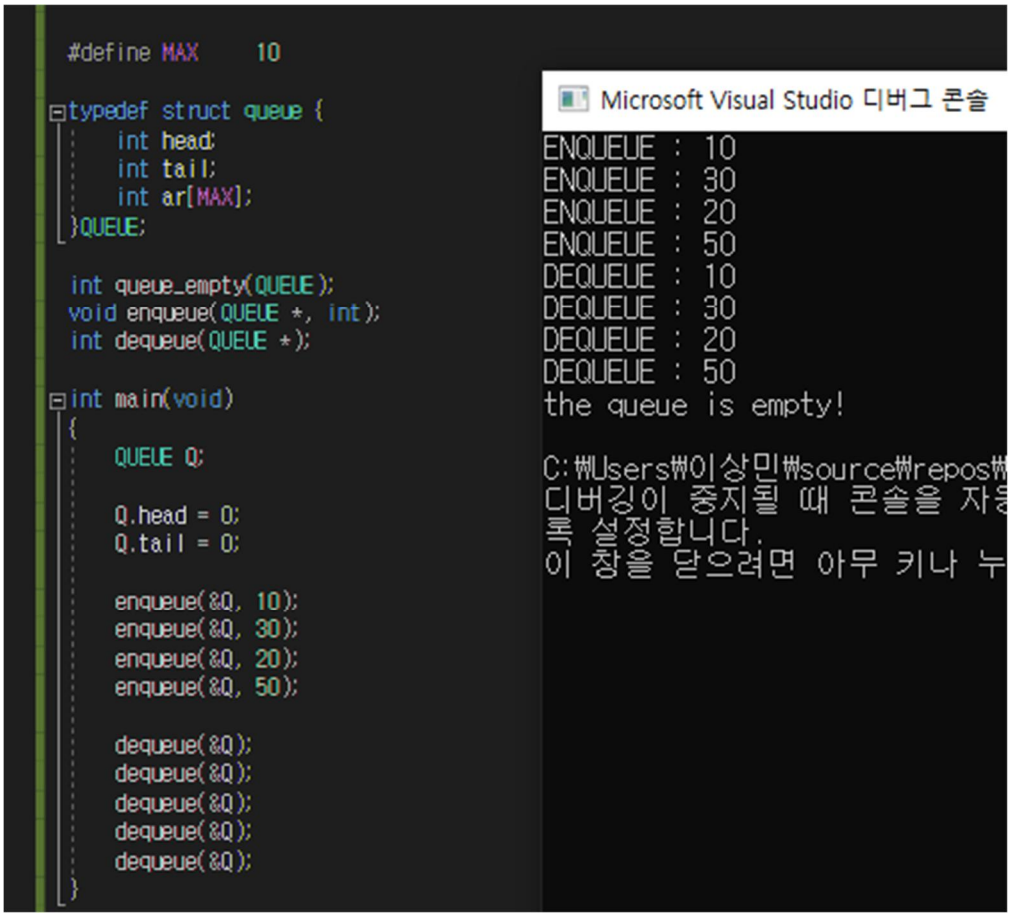
6. Write a program to create a queue using arrays which permits deletion from both the ends.
양쪽 끝에서 삭제할 수 있는 배열을 사용하여 대기열을 만드는 프로그램을 작성한다.

```
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 1

Enter the value to be added:10

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 1

Enter the value to be added:20

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 1

Enter the value to be added:30

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 4

The elements of the queue are : 102030
```

```
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 3

The element deleted is : 30
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 4

The elements of the queue are : 1020

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 2

The deleted element is : 10
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 3

The element deleted is : 20
INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 5

C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Pr
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [
이 창을 닫으려면 아무 키나 누르세요.
```

7. Write a program to create a queue using arrays which permits insertion and deletion at both the ends.

프로그램을 작성하여 양쪽 끝에 삽입과 삭제가 가능한 배열로 큐를 작성한다.

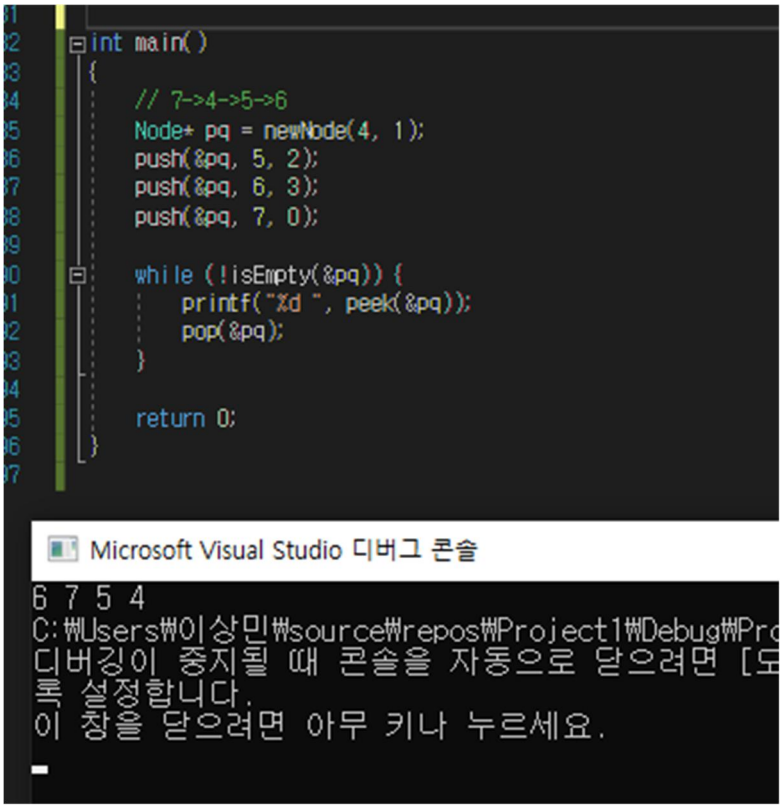


```
#define MAX     10

typedef struct queue {
    int head;
    int tail;
    int ar[MAX];
}QUEUE;

int queue_empty(QUEUE);
void enqueue(QUEUE *, int);
int dequeue(QUEUE *);

int main(void)
{
    QUEUE Q;

    Q.head = 0;
    Q.tail = 0;

    enqueue(&Q, 10);
    enqueue(&Q, 30);
    enqueue(&Q, 20);
    enqueue(&Q, 50);

    dequeue(&Q);
    dequeue(&Q);
    dequeue(&Q);
    dequeue(&Q);
    dequeue(&Q);
}
```

```
Microsoft Visual Studio 디버그 콘솔
ENQUEUE : 10
ENQUEUE : 30
ENQUEUE : 20
ENQUEUE : 50
DEQUEUE : 10
DEQUEUE : 30
DEQUEUE : 20
DEQUEUE : 50
the queue is empty!

C:₩Users₩이상민₩source₩repos₩
디버깅이 중지될 때 콘솔을 자
록 설정합니다.
이 창을 닫으려면 아무 키나 누
```

8. Write a program to implement a priority queue.

우선 순위 대기열을 구현하는 프로그램을 작성한다.



```
int main()
{
    // 7->4->5->6
    Node* pq = newNode(4, 1);
    push(&pq, 5, 2);
    push(&pq, 6, 3);
    push(&pq, 7, 0);

    while (!isEmpty(&pq)) {
        printf("%d ", peek(&pq));
        pop(&pq);
    }

    return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔
6 7 5 4
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Pro
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

9. Write a program to create a queue from a stack.

프로그램을 작성하여 스택에서 큐를 만드십시오.

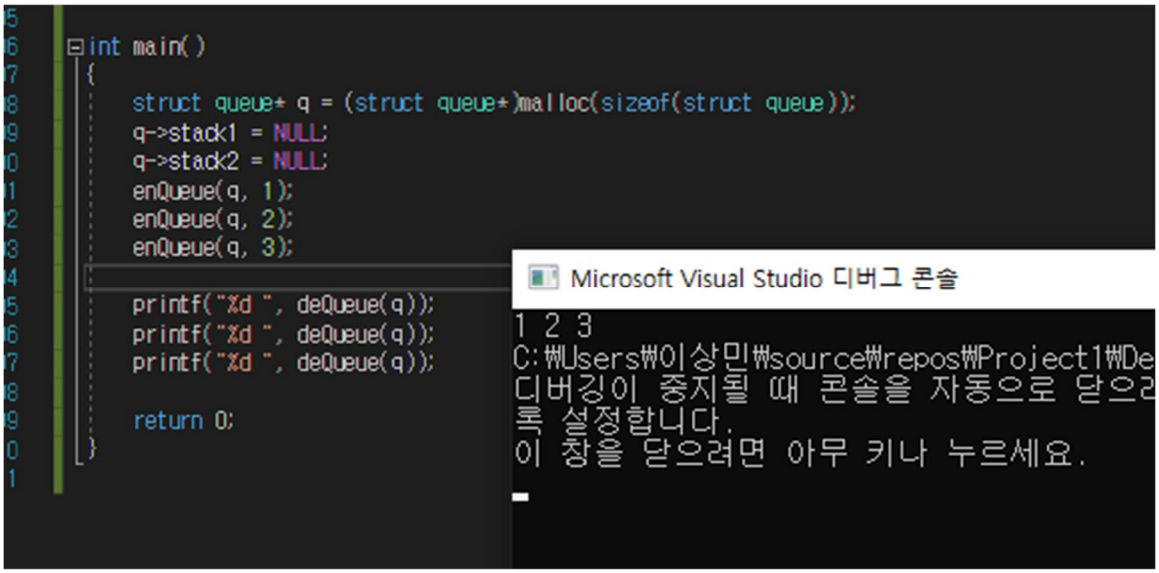
```c
struct Stack {
    int data;
    struct Stack *next;
};


struct Stack *CreateStack( ) {
    return NULL;
}

int isEmptyStack(struct Stack *top) {
    return (top == NULL);
}

void Push(struct Stack **top, int data) {
    struct Stack *newNode = (struct Stack*) malloc(sizeof(struct Stack));
    if (!newNode)
        return;
    newNode->data = data;
    newNode->next = *top;
    *top = newNode;
}
```

```c
int main( ) {
    struct Queue *Q = CreateQueue( );
    Q->S1 = Q->S2 = NULL;
    EnQueue(Q, 1);
    EnQueue(Q, 2);
    EnQueue(Q, 3);

    printf("%d ", DeQueue(Q));
    printf("%d ", DeQueue(Q));
    printf("%d ", DeQueue(Q));

    return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔
1 2 3
C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩
디버깅이 중지될 때 콘솔을 자동으로 닫으려면
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

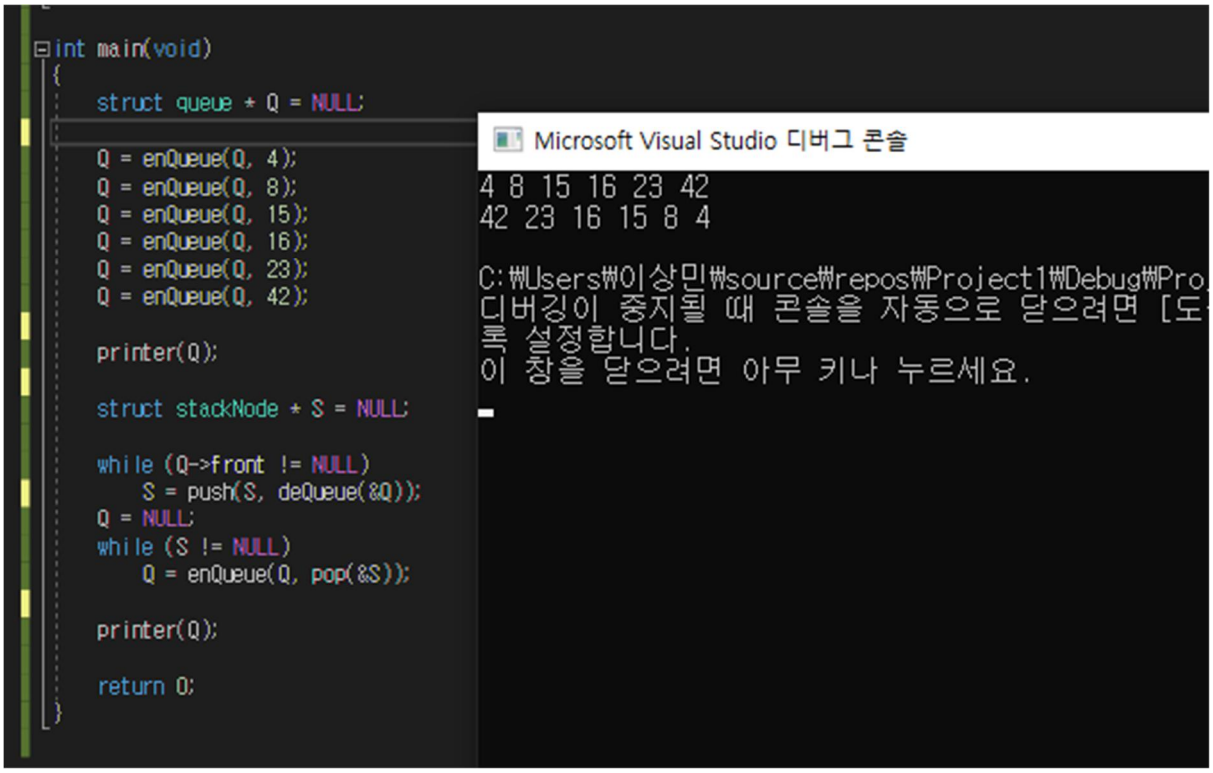
10. Write a program to create a stack from a queue.

대기열에서 스택을 생성하기 위한 프로그램을 작성한다.



11. Write a program to reverse the elements of a queue.

대기열의 요소를 반전시키는 프로그램을 작성한다.



```
int main(void)
{
    struct queue * Q = NULL;

    Q = enQueue(Q, 4);
    Q = enQueue(Q, 8);
    Q = enQueue(Q, 15);
    Q = enQueue(Q, 16);
    Q = enQueue(Q, 23);
    Q = enQueue(Q, 42);

    printer(Q);

    struct stackNode * S = NULL;

    while (Q->front != NULL)
        S = push(S, deQueue(&Q));
    Q = NULL;
    while (S != NULL)
        Q = enQueue(Q, pop(&S));

    printer(Q);

    return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔
4 8 15 16 23 42
42 23 16 15 8 4

C:₩Users₩이상민₩source₩repos₩Project1₩Debug₩Pro
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

12. Write a program to input two queues and compare their contents.

대기열 두 개를 입력하여 내용을 비교하는 프로그램을 작성한다.

```
37
38  // Driver Code
39  int main()
40  {
41      // Creating stacks
42      stack<string> stack1;
43      stack<string> stack2;
44
45      // Inserting elements to stack1
46      stack1.push("Geeks");
47      stack1.push("4");
48      stack1.push("Geeks");
49      stack1.push("Welcomes");
50      stack1.push("You");
51
52      // Inserting elements to stack2
53      stack2.push("Geeks");
54      stack2.push("4");
55      stack2.push("Geeks");
56      stack2.push("Welcomes");
57      stack2.push("You");
58
59      if (isSameStack(stack1, stack2))
60          cout << "Stacks are Same";
61      else
62          cout << "Stacks are not Same";
63
64      return 0;
65  }
```

Get URL

options | compilation | execution

```
Stacks are Same
```