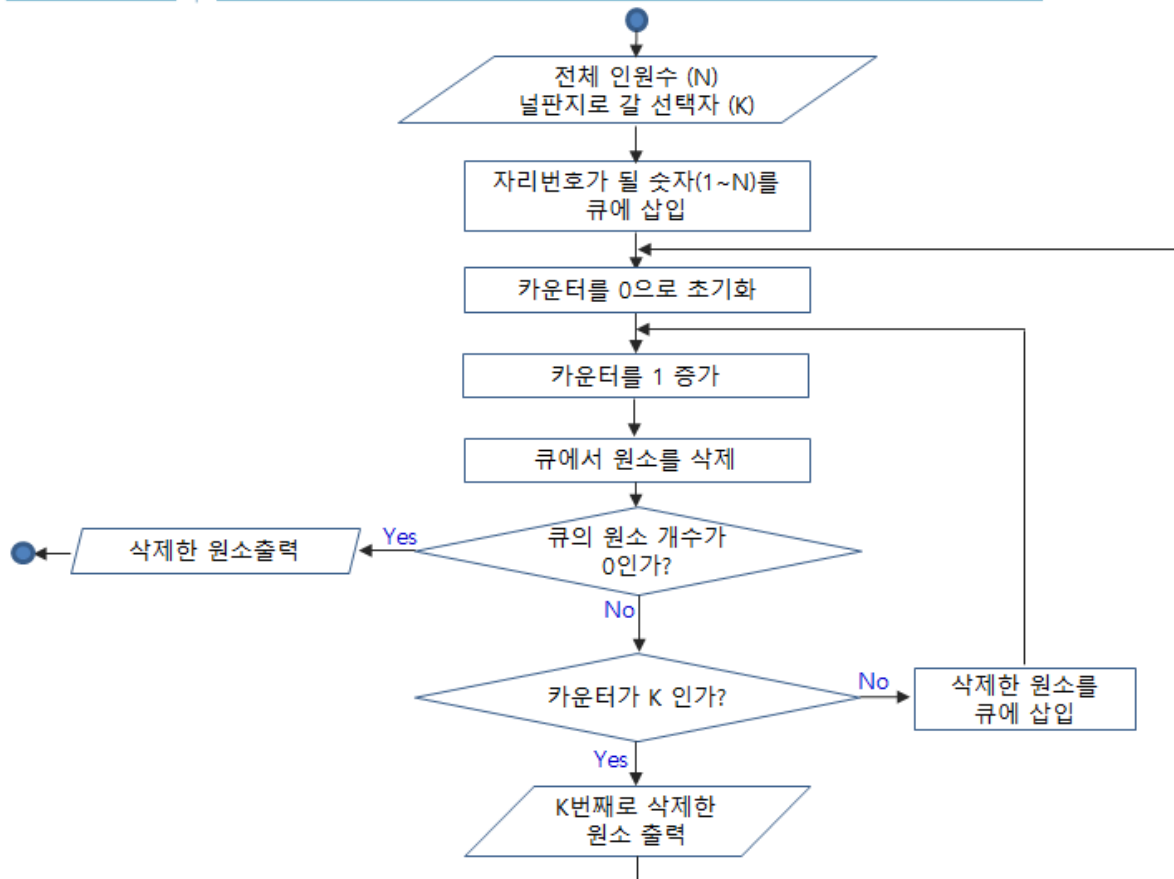


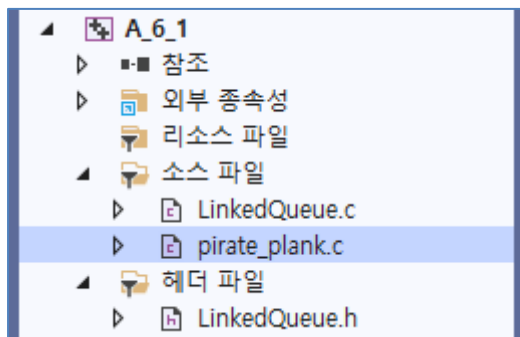
6장. 응용예제 답

응용예제
07

해적 널빤지 사형에서 살아남기



// 책 324쪽의 순서도를 위와 같이 수정해야 합니다.



예제 6-3의 연결큐를 이용하여 구현

LinkedQueue.h

```

//예제 6-3. LinkedQueue.h
#pragma once
typedef int element;    // 연결 큐 원소(element)의 자료형을 int로 수정!!

typedef struct QNode {   // 연결 큐의 노드를 구조체로 정의
    element data;
    struct QNode* link;
} QNode;

typedef struct {         // 연결 큐에서 사용하는 포인터 front와 rear를 구조체로 정의
    QNode* front, * rear;
} LQueueType;

LQueueType* createLinkedQueue(void);
int isLQEmpty(LQueueType* LQ);
void enLQueue(LQueueType* LQ, element item);
element deLQueue(LQueueType* LQ);
element peekLQ(LQueueType* LQ);
void printLQ(LQueueType* LQ);

```

LinkedQueue.c

```

//예제 6-3. LinkedQueue.c
#include <stdio.h>
#include <stdlib.h>
#include "LinkedQueue.h"

// 공백 연결 큐를 생성하는 연산
LQueueType* createLinkedQueue(void) {
    LQueueType* LQ;
    LQ = (LQueueType*)malloc(sizeof(LQueueType));
    LQ->front = NULL;
    LQ->rear = NULL;
    return LQ;
}

// 연결 큐가 공백 상태인지 검사하는 연산
int isLQEmpty(LQueueType* LQ) {
    if (LQ->front == NULL) {
        //printf(" Linked Queue is empty! ");
        return 1;
    }
    else return 0;
}

// 연결 큐의 rear에 원소를 삽입하는 연산
void enLQueue(LQueueType* LQ, element item) {
    QNode* newNode = (QNode*)malloc(sizeof(QNode));
    newNode->data = item;
    newNode->link = NULL;
    if (LQ->front == NULL) {           // 현재 연결 큐가 공백 상태인 경우

```

```

        LQ->front = newNode;
        LQ->rear = newNode;
    }
    else {                                     // 현재 연결 큐가 공백 상태가 아닌 경우
        LQ->rear->link = newNode;
        LQ->rear = newNode;
    }
}

// 연결 큐에서 원소를 삭제하고 반환하는 연산
element deLQueue(LQueueType* LQ) {
    QNode* old = LQ->front;
    element item;
    if (isLQEmpty(LQ)) return;
    else {
        item = old->data;
        LQ->front = LQ->front->link;
        if (LQ->front == NULL)
            LQ->rear = NULL;
        free(old);
        return item;
    }
}

// 연결 큐에서 원소를 검색하는 연산
element peekLQ(LQueueType* LQ) {
    element item;
    if (isLQEmpty(LQ)) return;
    else {
        item = LQ->front->data;
        return item;
    }
}

// 연결 큐의 원소를 출력하는 연산
void printLQ(LQueueType* LQ) {
    QNode* temp = LQ->front;
    printf(" Pirate's Plank : ["); //문제에 맞게 출력내용 수정!
    while (temp) {
        printf("%3d", temp->data); // element에 맞추어 출력형식을 %d로 수정 !!
        temp = temp->link;
    }
    printf(" ] ");
}

```

pirate_plank.c

//응용예제07. 해적 널판지 사형에서 살아남기

```
#define _CRT_SECURE_NO_WARNINGS //scanf 보안 경고 해제
#pragma warning(disable : 6031) //scanf의 반환값없음warning C6031을 해제하기 위해 추가.
#include<stdio.h>
#include "LinkedList.h"

int main(void) {
    int i, count=0, N, K, select;
    LQueueType* plank = createLinkedList(); // 연결 큐 생성

    //1) 인원수 N과 선택자 K 입력 받기
    printf("인원수 N: "); scanf("%d", &N);
    printf("선택자 K: "); scanf("%d", &K);

    //2) 자리번호 1~N을 큐에 삽입
    for (i = 0; i < N; i++)
        enqueue(plank, i + 1);
    //printf("Wn"); printLQ(plank); //현재 널판지에 있는 사람들 번호 확인용 출력

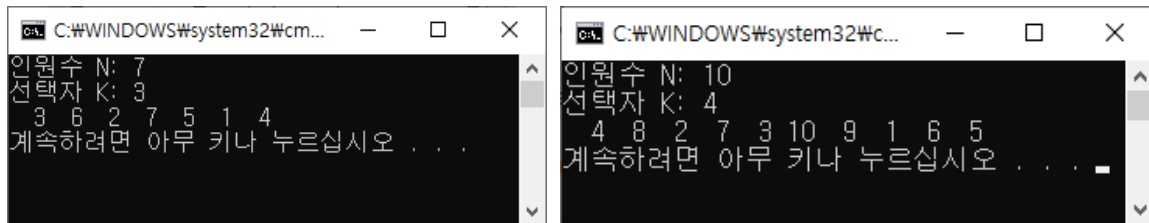
    //3) 해적 널판지 사형
    while(!isEmpty(plank)){
        count++;
        //널판지 맨 앞사람 찾기 (큐에 있는 맨 앞 번호) => select
        select = dequeue(plank);

        //select가 마지막 사람인 경우
        if (isEmpty(plank)) {
            printf("%3d", select); //마지막 살아남은 사람 출력
            break;
        }

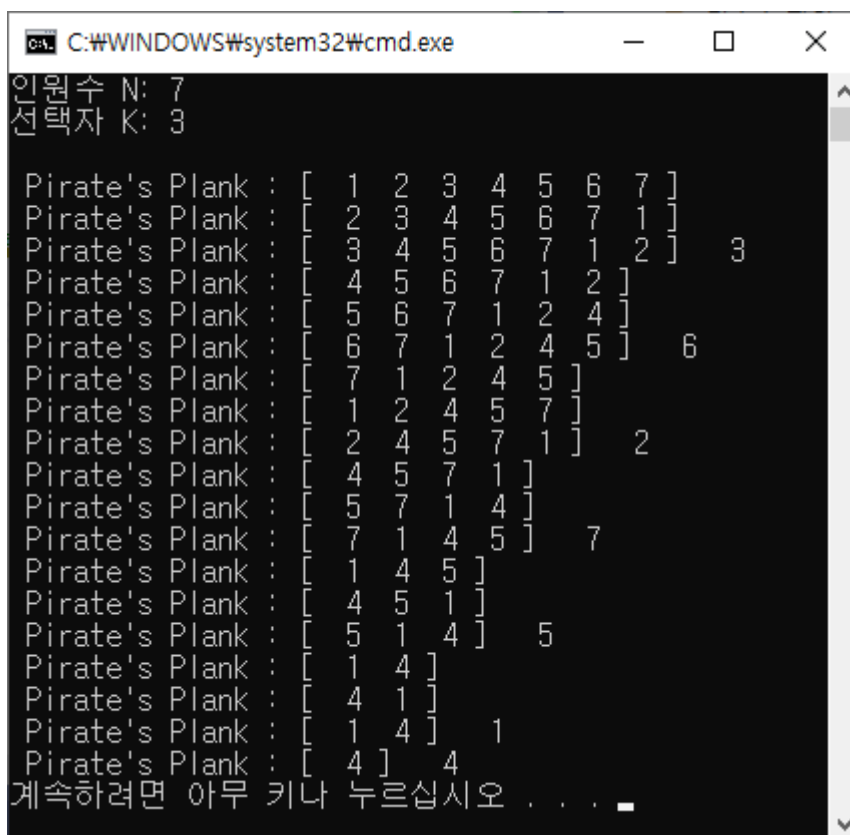
        //select가 널판지에서 떨어트릴 K번째 사람이면 출력
        if (count == K) {
            printf("%3d", select);
            count = 0; //카운트 초기화
        }
        else enqueue(plank, select); //아니면, 널판지 맨 뒤로 이동 (큐에 삽입)

        //printf("Wn"); printLQ(plank); //현재 널판지에 있는 사람들 확인용 출력
    }
    printf("Wn");
    return 0;
}
```

[실행 화면]

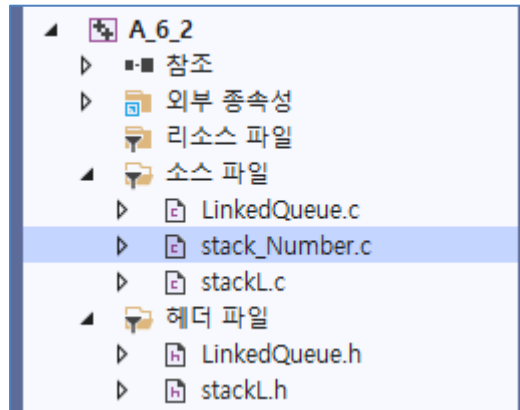


확인용 주석을 해제한 후, 실행결과 >



```
C:\WINDOWS\system32\cmd.exe
인원수 N: 10
선택자 K: 4

Pirate's Plank : [ 1 2 3 4 5 6 7 8 9 10 ]
Pirate's Plank : [ 2 3 4 5 6 7 8 9 10 1 ]
Pirate's Plank : [ 3 4 5 6 7 8 9 10 1 2 ]
Pirate's Plank : [ 4 5 6 7 8 9 10 1 2 3 ] 4
Pirate's Plank : [ 5 6 7 8 9 10 1 2 3 ]
Pirate's Plank : [ 6 7 8 9 10 1 2 3 5 ]
Pirate's Plank : [ 7 8 9 10 1 2 3 5 6 ]
Pirate's Plank : [ 8 9 10 1 2 3 5 6 7 ] 8
Pirate's Plank : [ 9 10 1 2 3 5 6 7 ]
Pirate's Plank : [ 10 1 2 3 5 6 7 9 ]
Pirate's Plank : [ 1 2 3 5 6 7 9 10 ]
Pirate's Plank : [ 2 3 5 6 7 9 10 1 ] 2
Pirate's Plank : [ 3 5 6 7 9 10 1 ]
Pirate's Plank : [ 5 6 7 9 10 1 3 ]
Pirate's Plank : [ 6 7 9 10 1 3 5 ]
Pirate's Plank : [ 7 9 10 1 3 5 6 ] 7
Pirate's Plank : [ 9 10 1 3 5 6 ]
Pirate's Plank : [ 10 1 3 5 6 9 ]
Pirate's Plank : [ 1 3 5 6 9 10 ]
Pirate's Plank : [ 3 5 6 9 10 1 ] 3
Pirate's Plank : [ 5 6 9 10 1 ]
Pirate's Plank : [ 6 9 10 1 5 ]
Pirate's Plank : [ 9 10 1 5 6 ]
Pirate's Plank : [ 10 1 5 6 9 ] 10
Pirate's Plank : [ 1 5 6 9 ]
Pirate's Plank : [ 5 6 9 1 ]
Pirate's Plank : [ 6 9 1 5 ]
Pirate's Plank : [ 9 1 5 6 ] 9
Pirate's Plank : [ 1 5 6 ]
Pirate's Plank : [ 5 6 1 ]
Pirate's Plank : [ 6 1 5 ]
Pirate's Plank : [ 1 5 6 ] 1
Pirate's Plank : [ 5 6 ]
Pirate's Plank : [ 6 5 ]
Pirate's Plank : [ 5 6 ]
Pirate's Plank : [ 6 5 ] 6
Pirate's Plank : [ 5 ] 5
계속하려면 아무 키나 누르십시오 . . .
```



예제 5-2의 연결 스택과 예제 6-3의 연결 큐를 사용하여 구현.

stack_Number.c

```
//응용예제08. 스택 수 판별하기
#define _CRT_SECURE_NO_WARNINGS //scanf 보안 경고 해제
#pragma warning(disable : 6031) //scanf의 반환값없음warning C6031을 해제하기 위해 추가.
#include<stdio.h>
#include "LinkedQueue.h"
#include "stackL.h"

#define MAX 20

int main(void) {
    int i, N, numbers[MAX], end=0;

    LQueueType* RQ = createLinkedQueue(); // 연결 큐 RQ 생성
    LQueueType* LQ = createLinkedQueue(); // 연결 큐 LQ 생성
    extern stackNode* top; //스택 MS의 top

    //1) 데이터크기 N과 테스트할 수열 numbers 입력
    scanf("%d", &N);
    for (i = 0; i < N; i++)
        scanf("%d", &numbers[i]);

    //2) 1~N으로 RQ 초기화
    for (i= 0; i < N; i++)
        enLQueue(RQ, i+1); //1~N으로 RQ 초기화

    //3) 스택 수 판별
    i = 0;
    while(i < N) {
```

```

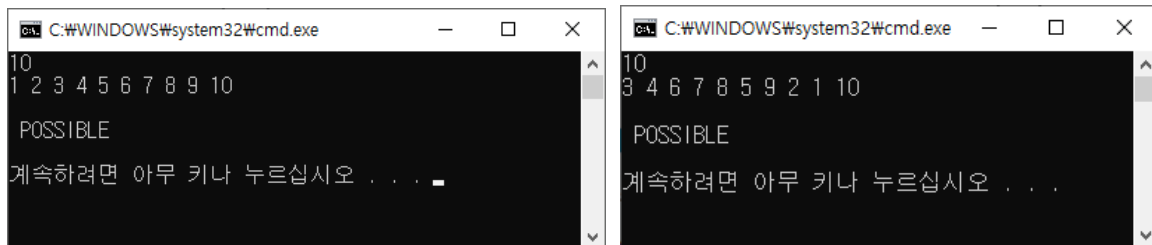
        if (peekLQ(RQ) == numbers[i]) {
            enLQueue(LQ, deLQueue(RQ));
            i++; // numbers의 다음 검사를 위해 인덱스 i값 증가
        }
        else if (peekLQ(RQ) > numbers[i])
            if ((top->data) == numbers[i]) {
                enLQueue(LQ, pop());
                i++; // numbers의 다음 검사를 위해 인덱스 i값 증가
            }
            else {
                printf("Wn IMPOSSIBLEWnWn");
                end = 1; //작업 완료 태그 설정
                break;
            }
        else push(deLQueue(RQ));
    }

    if(isStackEmpty() && !end) printf("Wn POSSIBLEWnWn");
    else if (!isStackEmpty() && !end) printf("Wn IMPOSSIBLEWnWn");

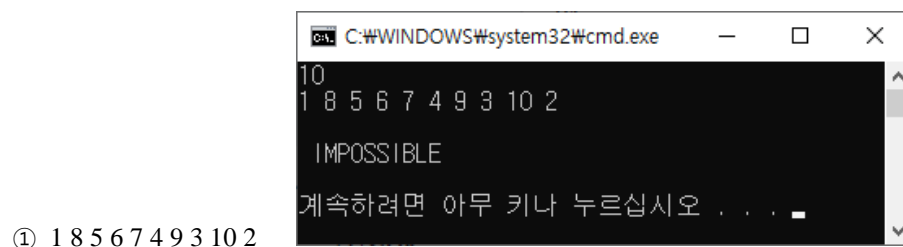
    return 0;
}

```

실행 결과 >



6장 연습문제 6번 보기 ①~④에 대한 실행 결과 >



① 1 8 5 6 7 4 9 3 10 2

② 3 4 6 7 8 5 9 2 1 10

```
C:\WINDOWS\system32\cmd.exe
10
3 4 6 7 8 5 9 2 1 10
POSSIBLE
계속하려면 아무 키나 누르십시오 . . .
```

③ 3 4 8 5 6 7 2 9 1 10

```
C:\WINDOWS\system32\cmd.exe
10
3 4 8 5 6 7 2 9 1 10
IMPOSSIBLE
계속하려면 아무 키나 누르십시오 . . .
```

④ 9 4 7 5 6 3 8 1 2 10

```
C:\WINDOWS\system32\cmd.exe
10
9 4 7 5 6 3 8 1 2 10
IMPOSSIBLE
계속하려면 아무 키나 누르십시오 . . .
```