

# Week1\_스터디

[통계](#) [수정](#) [삭제](#)

yuyu7123 · 1일 전 · 비공개

 0

10828   2346   스택   원형 큐

## Smarcle\_23Algorithm



▼ 목록 보기

2/3



## Baekjoon\_2346 : 풍선 터뜨리기

원형으로 놓인 풍선 -&gt; 원형 큐를 사용

### 풀이 과정

1. 원형 큐에 풍선 순서와 안의 값 전달해 노드 생성
2. 현재 맨 앞의 풍선의 순서 출력
3. 풍선 안의 값 변수에 저장
4. 풍선을 터뜨리는 노드 삭제 연산
5. 앞서 저장한 변수 값을 사용하여 풍선 이동
6. 2~5과정 큐가 공백이 되기 전까지 반복

### 문제 해결

- 풍선의 순서 출력과 이동 제거를 한 함수에 구현하려고 했지만 실패  
따로 함수 생성
- 순서 출력에서 몇 개의 값만 다른 오류가 생김  
풍선 이동 부분에서 다음 풍선의 값을 전달하여 잘못 이동하는 오류 -> 삭제 전 풍선 안의 값 저장하는 변수로 해결

## 코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

// 구조체 정의
typedef struct Balloon {
    int order;
    int value;
    struct Balloon* next;
    struct Balloon* prev;
} Balloon;

typedef struct {
    Balloon* front;
    Balloon* rear;
    int size;
} CircularQueue;

// 원형 큐 초기화 함수
void initQueue(CircularQueue* queue) {
    queue->front = NULL;
    queue->rear = NULL;
    queue->size = 0;
}

// 풍선 추가
void enqueue(CircularQueue* queue, int order, int value) {
    Balloon* newBalloon = (Balloon*)malloc(sizeof(Balloon));
    newBalloon->order = order;
    newBalloon->value = value;

    if (queue->size == 0) {
        newBalloon->next = newBalloon;
        newBalloon->prev = newBalloon;
        queue->front = newBalloon;
        queue->rear = newBalloon;
    }
    else {
        newBalloon->next = queue->front;
        newBalloon->prev = queue->rear;
        queue->rear->next = newBalloon;
        queue->front->prev = newBalloon;
        queue->rear = newBalloon;
    }

    queue->size++;
}

// 풍선 이동
void moveBalloon(CircularQueue* queue, int count) {
    if (count > 0) {
        for (int i = 1; i < count; i++) {
            queue->front = queue->front->next;
            queue->rear = queue->rear->next;
        }
    }
}
```

```

    }
    else {
        for (int i = 0; i < -count; i++) {
            queue->front = queue->front->prev;
            queue->rear = queue->rear->prev;
        }
    }
}

// 풍선 제거
void popBalloon(CircularQueue* queue, int* order, int* value) {
    if (queue->size == 0) {
        *order = -1;
        *value = -1;
        return;
    }

    *order = queue->front->order;
    *value = queue->front->value;

    Balloon* temp = queue->front;
    if (queue->size == 1) {
        queue->front = NULL;
        queue->rear = NULL;
    }
    else {
        queue->front = temp->next;
        queue->rear->next = queue->front;
        queue->front->prev = queue->rear;
    }

    free(temp);
    queue->size--;
}

int main() {
    int N, n, m;
    scanf("%d", &N);

    CircularQueue queue;
    initQueue(&queue);

    for (int i = 0; i < N; i++) {
        scanf("%d", &n);
        enqueue(&queue, i, n);
    }

    int order, value;

    while (queue.size > 0) {
        printf("%d ", queue.front->order+1);
        m = queue.front->value;
        popBalloon(&queue, &order, &value);
        if (queue.size == 0) break;
        moveBalloon(&queue, m);
    }
}

```

```

    return 0;
}

```

## Baekjoon\_10828 : 스택

기본 스택 구현, 문자열 비교 사용

### 풀이 과정

1. 스택의 push, pop, size, empty, top 연산 구현
2. 명령을 수행하기 위해 문자열 비교 함수 사용

### 문제 해결

- 제출시 컴파일 에러  
empty함수의 위에 선언된 함수에서 호출문제로 empty함수의 위치 변경으로 해결
- 제출시 시간초과 에러  
문자열 크기 조절 -> 해결X  
함수 내에 있던 출력문 return으로 변경한 후 main에서 출력 -> 해결X  
스택 사이즈 100에서 10000으로 증가 -> 해결

### 코드

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_STACK_SIZE 10000

// 스택 요소의 자료형 정의
typedef int element;

element data[MAX_STACK_SIZE];
int top;

// 스택 초기화
void init_stack() { top = -1; }

// 스택이 비어있으면 1, 아니면 0 반환
int empty() {
    return (top == -1);
}

// 정수 e를 스택에 넣는 연산
void pushX(element e) {
    data[++top] = e;
}

```

```
}

// 가장 위에 있는 원소 반환, 제거
int pop() {
    if (empty()) return -1;
    else return data[top--];
}

// 스택에 있는 원소 수 반환
int size() {
    if (empty()) return 0;
    else return (top + 1);
}

// 가장 위에 있는 원소 반환
int peek() {
    if (empty()) return -1;
    else return data[top];
}

int main() {

    int n, m;
    char ch[7];
    init_stack();
    scanf("%d", &n);
    getchar();

    for (int i = 0; i < n; i++) {
        scanf("%s", ch);
        if (strcmp(ch, "push") == 0) {
            scanf("%d", &m);
            pushX(m);
        }
        else if (strcmp(ch, "pop") == 0) {
            printf("%d\n", pop());
        }
        else if (strcmp(ch, "size") == 0) {
            printf("%d\n", size());
        }
        else if (strcmp(ch, "empty") == 0) {
            printf("%d\n", empty());
        }
        else if (strcmp(ch, "top") == 0) {
            printf("%d\n", peek());
        }
    }

    return 0;
}
```



유정

다음 포스트  
Week2\_과제



이전 포스트  
Week1\_과제

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by  
Stellate