

1주차 자료구조 복습

[통계](#) [수정](#) [삭제](#)

iming03 · 방금 전 · 비공개

 0[스택](#) [알고리즘 기초](#) [연결리스트](#) [자료구조](#) [큐](#) [트리](#)

알고리즘 스터디

[▼ 목록 보기](#)

1/1



#1 10845번 큐

이 문제는 큐의 기본적인 메소드들을 사용해 큐를 구현하는 문제이다.
사용할 함수들은 문제에 나온 함수와 같다.

push X: 정수 X를 큐에 넣는 연산이다.

pop: 큐에서 가장 앞에 있는 정수를 빼고, 그 수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력

size: 큐에 들어있는 정수의 개수를 출력한다.

empty: 큐가 비어있으면 1, 아니면 0을 출력한다.

front: 큐의 가장 앞에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.

back: 큐의 가장 뒤에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.`

문제를 풀기 전 큐에 대해 먼저 알아보겠다.

큐(QUEUE)는 기본적인 자료구조 중 한 가지로 먼저 집어 넣은 데이터가 먼저 나오는 **FIFO(First In First Out)**구조로 저장하는 형식을 말한다.

삽입은 큐의 **후단(rear)**에서, 삭제는 **전단(front)**에서 이루어진다.

[코드]

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

int queue[10001];
int queue_size = 0;

void push(int x){
    queue[queue_size] = x;
    queue_size += 1;
}

int empty() {
    if(queue_size == 0){
        return 1;
    }
    return 0;
}

int pop() {
    if(empty()) {
        return -1;
    }
    queue_size -= 1;
    return queue[0];
}

int front(){
    if(empty()) {
        return -1;
    }
    return queue[queue_size - queue_size];
}

int back() {
    if(empty()) {
        return -1;
    }
    return queue[queue_size-1];
}

void setting(){
    for(int i = 0; i < queue_size; i++)
    {
        queue[i]=queue[i+1];
    }
}

int main() {
```

```

int N = 0, x=0;
char arr[6] = {0, };

scanf("%d", &N);
for(int i = 0; i < N; i++) {
    scanf("%s", arr);

    if(!strcmp(arr, "push")){
        scanf("%d", &x);
        push(x);
    }
    else if(!strcmp(arr, "pop")){
        printf("%d\n", pop());
        setting();
    }
    else if(!strcmp(arr, "empty")) {
        printf("%d\n", empty());
    }
    else if(!strcmp(arr, "size")) {
        printf("%d\n", queue_size);
    }
    else if(!strcmp(arr, "front")){
        printf("%d\n", front());
    }
    else if(!strcmp(arr, "back")) {
        printf("%d\n", back());
    }
}

return 0;
}

```

오랜만에 하려니까 정말 ㅎㅎ... 하나도 생각 안나서 작년에 공부했던 자료를 찾아봤다... 역시 공부는 꾸준히 하는 거구나,,^^ 다들 화이팅!!

#2 10828번 스택

이 문제는 스택의 기본적인 메소드들을 사용해 스택을 구현하는 문제이다. 사용할 함수들은 문제에 나온 함수와 같다.

이 문제는 위에 푼 큐 문제와 매우 비슷하다!!

push X: 정수 X를 스택에 넣는 연산이다.

pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을

size: 스택에 들어있는 정수의 개수를 출력한다.

empty: 스택이 비어있으면 1, 아니면 0을 출력한다.

top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

문제를 풀기 전 스택에 대해 먼저 알아보겠다.

스택(STACK)은 기본적인 자료구조 중 한 가지로 가장 나중에 집어 넣은 데이터가 가장 먼저 나오는 **LIFO(Last In First Out)**구조로 저장하는 형식을 말한다. (큐와 나오는 순서 반대)

스택에 저장된 원소는 **top**으로 정한 곳에서만 접근 가능하다.

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

int stack[10001];
int count = -1;

int empty() {
    if (count == -1)
        return 1;
    else
        return 0;
}
int top() {
    if (empty())
        return -1;
    else
        return stack[count];
}
int push(int x) {
    stack[++count] = x;
}
int pop() {
    if (empty())
        return -1;
    else
        return stack[count--];
}
int size() {
    return count+1;
}
int main(void) {
    int N;
    char arr[6];

    scanf("%d", &N);

    for (int i = 0; i < N; i++) {
        scanf("%s", arr);
        if (strcmp(arr, "push") == 0) {
            int num;
            scanf("%d", &num);
            push(num);
        }
    }
}
```

```
    else if (strcmp(arr, "pop") == 0) {  
        printf("%d\n", pop());  
    }  
    else if (strcmp(arr, "top") == 0) {  
        printf("%d\n", top());  
    }  
    else if (strcmp(arr, "size") == 0) {  
        printf("%d\n", size());  
    }  
    else if (strcmp(arr, "empty") == 0) {  
        printf("%d\n", empty());  
    }  
}  
return 0;  
}
```



강민서

about computer

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate