

컴퓨터 프로그래밍 (Computer Programming)

이 선 순



10. 배열



목차

1. 배열의 이해
2. 2차원 배열
3. 배열의 활용 : 스택

03

배열의 활용 : 스택

03 배열의 활용 : 스택

■ 스택의 개념

- 한쪽 끝이 막힌 터널과 같은 자료 구조, 가장 나중에 들어간 것이 가장 먼저 나오는 LIFO 구조
- 스택은 JAVA뿐만 아니라 다른 프로그래밍 언어에서도 많이 활용되는 구조임

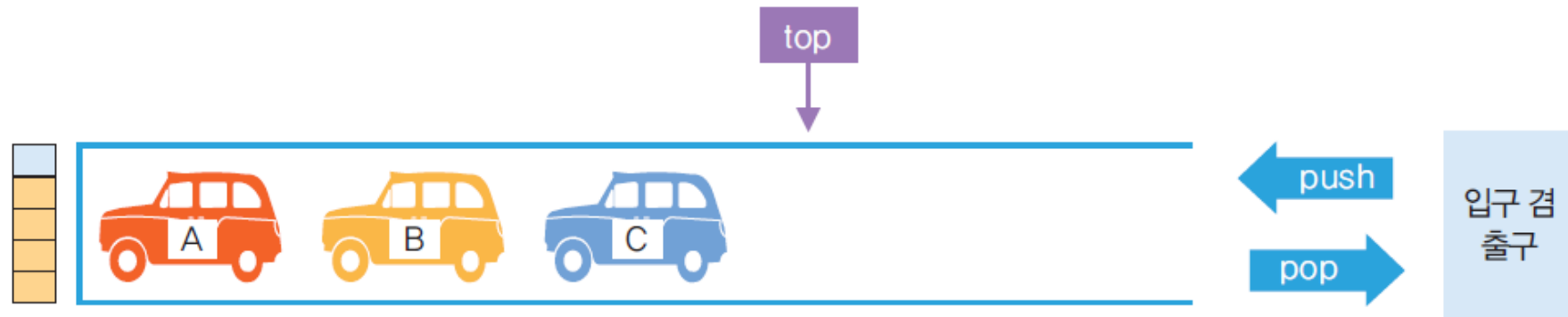


그림 8-21 스택의 기본 개념

- 들어가는 순서 : $A \rightarrow B \rightarrow C$, 나오는 순서 : $C \rightarrow B \rightarrow A$
- 터널 안에 자동차가 3대 들어갔는데 가장 마지막에 들어간 자동차 C의 다음을 top라고 함
 - 자동차 C가 나가면 top의 위치는 자동차 B의 다음으로 바뀜
 - top 은 현재 스택 안에 들어있는 데이터 중 가장 마지막 데이터의 다음을 가리킴
- 데이터를 넣는 것을 push, 데이터를 빼는 것을 pop

03 배열의 활용 : 스택

■ 배열로 스택만들기

■ 초기화

```
char[] stack= new char[5];  
int top=0;
```

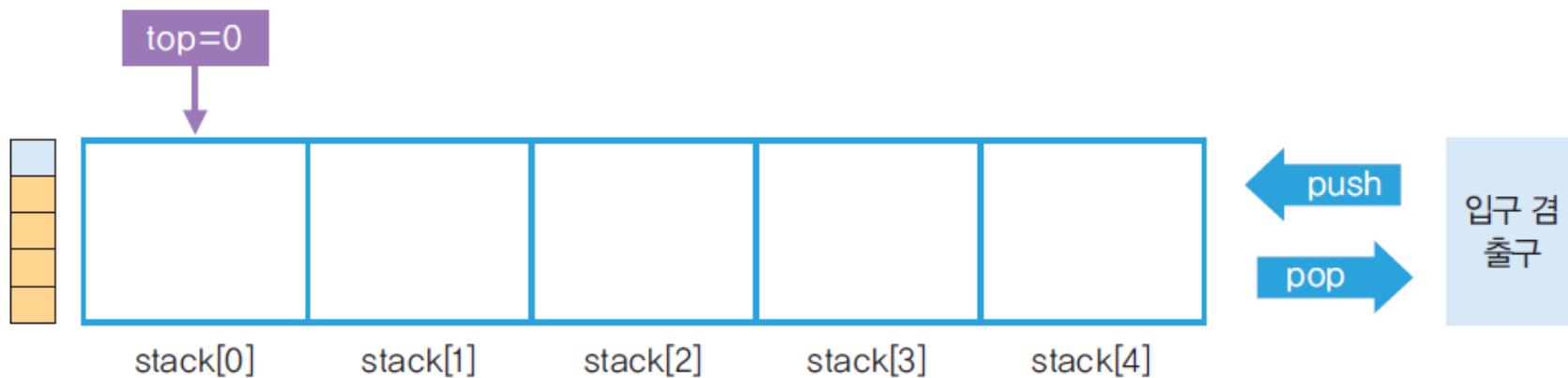


그림 8-22 비어 있는 터널

- 이 배열을 막힌 터널(스택)이라 생각한다면 현재 자동차가 없으므로 `top`은 0을 가리킴

03 배열의 활용 : 스택

■ 배열로 스택만들기

- 자동차 1대 넣기(push)

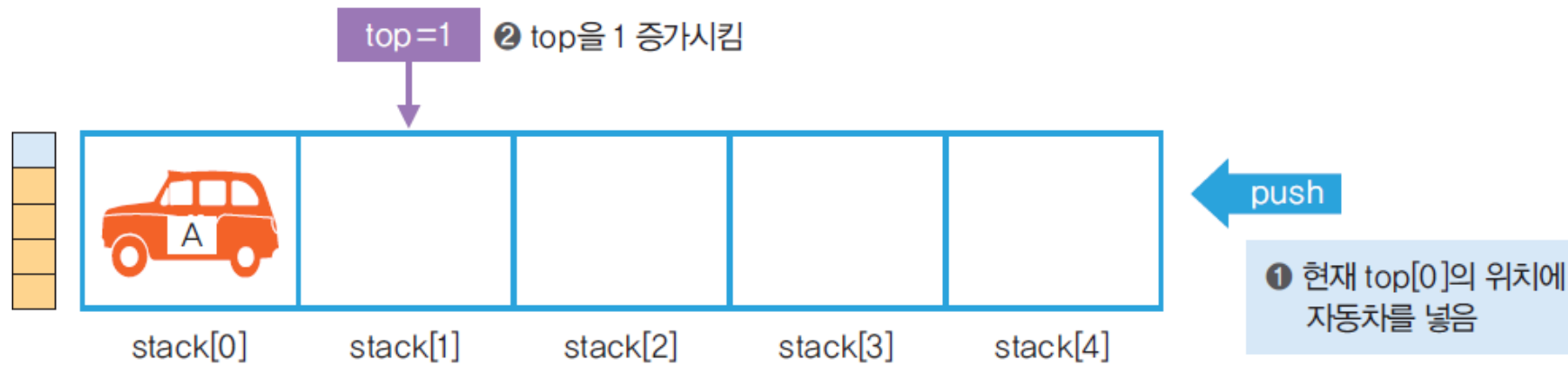


그림 8-23 터널에 자동차 1대 넣기

- 자동차 A를 넣으면 `top`은 0에서 1로 바뀌고, 위치는 `stack[0]`에서 `stack[1]`로 이동함

03 배열의 활용 : 스택

■ 배열로 스택만들기

- 자동차 B,C 터널에 넣기

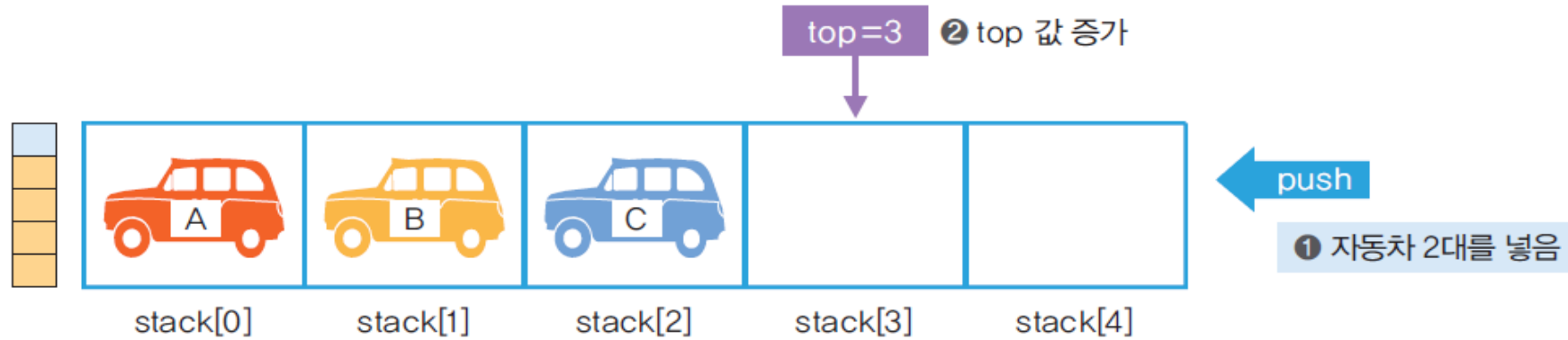


그림 8-24 터널에 자동차 3대 넣기

- 같은 방식으로 자동차 B와 자동차 C를 터널에 넣으면 `top`은 3이 되어, 위치는 `stack[3]`의 위치로 이동함

03 배열의 활용 : 스택

■ 배열로 스택만들기

- 자동차 1대 빼기

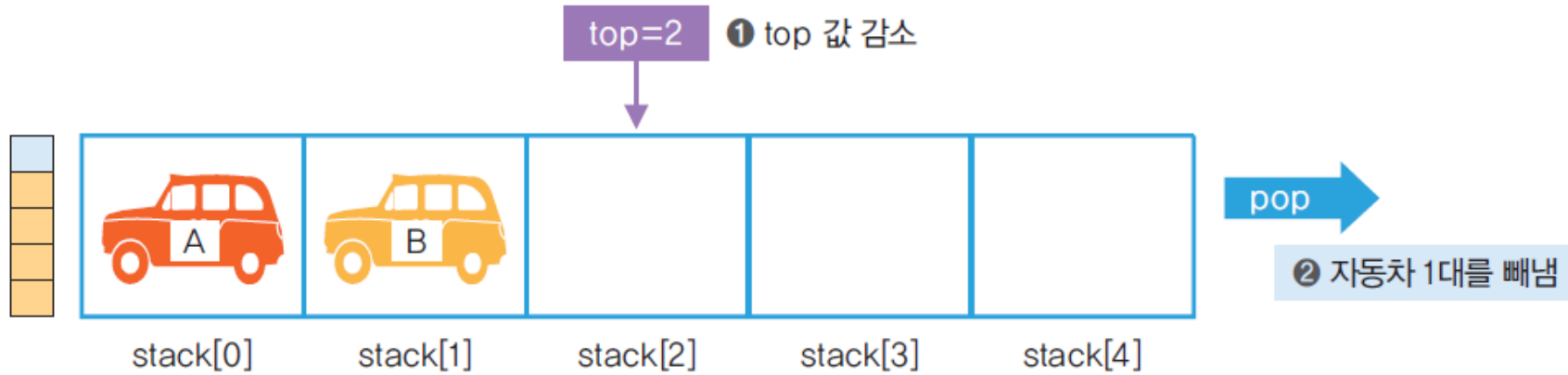


그림 8-25 터널에서 자동차 1대 빼기

- 자동차 1대를 빼면 `top`을 1 감소시킨 후 그 자리의 자동차를 빼내면 됨

03 배열의 활용 : 스택

■ 배열로 스택만들기

■ 실습8-11 스택구현1

```
1 public class Ex08_11 {  
2     public static void main(String[] args) {  
3         char[] stack = new char[5];  
4         int top = 0;  
5  
6         stack[top] = 'A';  
7         System.out.printf(" %c 자동차가 터널에 들어감\n", stack[top]);  
8         top++;  
9  
10        stack[top] = 'B';  
11        System.out.printf(" %c 자동차가 터널에 들어감\n", stack[top]);  
12        top++;  
13  
14        stack[top] = 'C';  
15        System.out.printf(" %c 자동차가 터널에 들어감\n", stack[top]);  
16        top++;  
17  
18        System.out.printf("\n");  
19    }  
}
```

스택을 위한 자료구조를 배열로 선언
top의 초기값 0을 선언

6행 : 먼저 현재의 스택 **top**위치에 자동차 **A**를 넣음

8행 : **top**값을 1 증가시킴

03 배열의 활용 : 스택

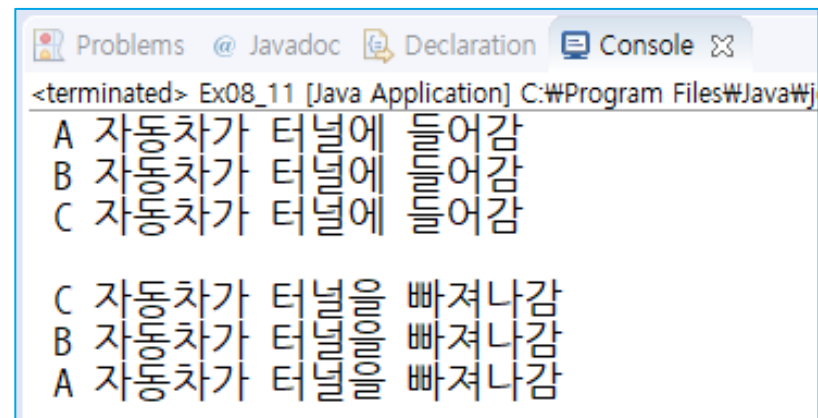
■ 배열로 스택만들기

■ 실습8-11 스택구현1

```
20     top--;  
21     System.out.printf(" %c 자동차가 터널을 빠져나감\n", stack[top]);  
22     stack[top] = ' ';  
23  
24     top--;  
25     System.out.printf(" %c 자동차가 터널을 빠져나감\n", stack[top]);  
26     stack[top] = ' ';  
27  
28     top--;  
29     System.out.printf(" %c 자동차가 터널을 빠져나감\n", stack[top]);  
30     stack[top] = ' ';  
31  
32 }  
33 }
```

터널에서 자동차를 빼내는 과정
먼저 현재의 **top**값 3에서 1을 감소시키고
22행에서 자동차 C자리에 빈칸을 넣으면 자동차를
빼내는 효과가 나타남

Top의 값을 계속 1씩 줄임



```
Problems @ Javadoc Declaration Console ⌵  
<terminated> Ex08_11 [Java Application] C:\Program Files\Java\j  
A 자동차가 터널에 들어감  
B 자동차가 터널에 들어감  
C 자동차가 터널에 들어감  
  
C 자동차가 터널을 빠져나감  
B 자동차가 터널을 빠져나감  
A 자동차가 터널을 빠져나감
```

03 배열의 활용 : 스택

- top이 0일 때 자동차를 빼내라는 명령이 들어오거나 top이 5일 때 자동차를 넣으라는 명령의 경우 오류 발생. 따라서 다음과 같이 수정해줘야 함

6~8행 수정 (10~12행, 14~16행도 수정)

```
if (top >= 5) {  
    printf("터널이 꽉 차서 차가 못 들어감.\n");  
}  
else {  
    stack[top] = 'A';  
    System.out.printf(" %c 자동차가 터널에 들어감\n", stack[top]);  
    top ++;  
}
```

03 배열의 활용 : 스택

자동차가 빠져나가는 20~22행 수정 (24~26행, 28~30행도 수정)

```
if (top <= 0) {  
    printf("현재 터널에 자동차가 없음\n");  
}  
else {  
    top --;  
    System.out.printf ("%c 자동차가 터널을 빠져나감\n", stack[top]);  
    stack[top] = ' ';  
}
```

03 배열의 활용 : 스택

■ 배열로 스택만들기

- 실습8-12 스택구현2
- 실습8-11을 수정하여 1을 입력하면 자동차가 들어가고, 2를 입력하면 자동차가 빠져나가도록 프로그램을 작성
- 입력되는 자동차는 A, B, C, ...순으로 증가함

```
1 import java.util.Scanner;
2
3 public class Ex08_12 {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         char[] stack = new char[5]; 5대의 자동차가 들어갈 수 있는 막힌 터널(스택)을 정의함
7         int top = 0; 초기에는 1대도 없으므로 7행에서 top을 0으로 초기화함
8
9         char carName = 'A'; 터널에 들어가는 자동차 이름은 A부터 시작함
10        int select = 9; 사용자가 입력할 변수인 select를 선언- 1: 자동차 넣기, 2:자동차 빼기, 3: 끝내기
11
12        while (select != 3) { 사용자의 선택이 3이 아니면 while문을 반복함
13            System.out.printf("<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : ");
14            select = s.nextInt();
15            사용자가 선택하는 값
```

- Select 를 9로 초기화한 이유 : 12행의 while문을 무조건 실행하기 위해서인데 9가 아니더라도 3이 아니라면 어떤 숫자든 상관없음. 어차피 14행에서 새로 입력받음

03 배열의 활용 : 스택

■ 배열로 스택만들기

■ 실습8-12 스택구현2

```
16      switch (select) {
17      case 1:
18          if (top >= 5) {
19              System.out.printf("터널이 꽉 차서 차가 못 들어감\n");
20          } else {
21              stack[top] = carName++;
22              System.out.printf(" %c 자동차가 터널에 들어감\n", stack[top]);
23              top++;
24          }
25          break;
26
```

터널에 자동차가 5대 있으면 더 이상 못 들어감

사용자가 1(넣기)을 선택하면 실행함

빈 곳이 있으면(자동차가 5대 미만) 자동차를 넣고 top1을 증가시킴

① 자동차 넣기

- 17~25행 처리. 18행에서 자동차를 넣는데, top이 5라면(즉 스택이 꽉 차 있다면 19행에서 더 이상 자동차가 들어가지 못한다는 메시지를 출력하고 25행으로 감.
- 25행의 break 문에 의해 44행의 밖으로 나간 후 다시 12행의 while 문으로 이동. 만약 top이 5 미만이면 21~23행을 수행. 21행에서 스택에 현재의 자동차(처음에는 A)를 넣은 다음 자동차의 이름을 증가시킴(처음이 A이므로 B가 됨). 그리고 22, 23행에서 들어간 자동차의 이름을 출력한 다음 top을 1 증가시킴.
- 25행을 실행하여 switch 문을 빠져나감

03 배열의 활용 : 스택

■ 배열로 스택만들기

■ 실습8-12 스택구현2

터널에 자동차가 1대도 없으면 빼낼 자동차가 없다

```
27 case 2:
28     if (top <= 0) {
29         System.out.printf("빠져나갈 자동차가 없음\n");
30     } else {
31         top--;
32         System.out.printf(" %c 자동차가 터널에서 빠짐\n", stack[top]);
33         stack[top] = ' ';
34     }
35     break;
36
```

사용자가 2(빼기)를
선택하면 실행함

빼낼 자동차가 있으면(1대 이상) top 1을 감소시키고 자동차를 빼낸다.
그리고 빠져나간 자리를 빈칸으로 채운다

② 자동차 빼기

- 27~35행 처리. 28행에서 자동차를 빼내야 하는데, top이 0이라면(자동차가 1대도 없다면) 29행에서 더 이상 빼낼 자동차가 없다는 메시지를 출력하고 35행 이동.
- 35행의 break 문에 의해 44행의 밖으로 나간 후 다시 12행의 while 문으로 이동. 만약 top이 0보다 크면 31~33행을 수행. 31행에서 빼낼 자동차의 위치는 현재의 top보다 한 칸 아래에 있기 때문에 먼저 top을 1 감소. 32, 33행에서 빠져나간 자동차의 이름을 출력한 다음 자동차가 빠진 위치에 공백(' ')을 채움.
- 마찬가지로 35행 실행하여 switch 문을 빠져나감

03 배열의 활용 : 스택

■ 배열로 스택만들기

■ 실습8-12 스택구현2

```
37
38
39
40
41
42
43
44
45
46
47
48
49
50
```

```
case 3:
    System.out.printf("현재 터널에 %d대가 있음.\n", top);
    System.out.printf("프로그램을 종료합니다.\n");
    break;

default:
    System.out.printf("잘못 입력했습니다. 다시 입력하세요. \n");
}

}

s.close();

}
```

switch 문을 벗어난다.

사용자가 1,2,3외의 값을 입력하면 처리한다.

사용자가 3(끝)을 선택하면
현재의 자동차 대수를 출력하고
종료함

③ 끝

37~40행 처리. 38, 39행에서 현재 터널에 들어 있는 자동차의 대수(top)를 출력한 다음 프로그램을 종료한다는 메시지를 출력. 40행에 의해 switch 문을 빠져나간 후 12행을 다시 수행, select가 3이므로 while 문도 종료, 전체 프로그램을 빠져나감

④ 그 외의 값 : 그 외의 값을 입력하면 42행의 default: 부분이 수행. 43행에서 잘못 입력했다는 메시지를 출력하고 다시 12행 이동. select가 3이 아니므로 다시 13행부터 수행.

03 배열의 활용 : 스택

■ 배열로 스택만들기

■ 실습8-12 스택구현2

```
Problems @ Javadoc Declaration Console
<terminated> Ex08_12 [Java Application] C:\Program Files\Java\jdk-11.0.6\bin
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
A 자동차가 터널에 들어감
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
B 자동차가 터널에 들어감
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
C 자동차가 터널에 들어감
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
D 자동차가 터널에 들어감
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
E 자동차가 터널에 들어감
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
터널이 꽉 차서 차가 못 들어감
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
E 자동차가 터널에서 빠짐
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
D 자동차가 터널에서 빠짐
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
C 자동차가 터널에서 빠짐
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
B 자동차가 터널에서 빠짐
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
A 자동차가 터널에서 빠짐
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
빠져나갈 자동차가 없음
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 3
현재 터널에 0대가 있음.
프로그램을 종료합니다.
```

Self study 10-3

1. [실습8-11]을 바탕으로 top이 0 또는 5일때 오류처리가 되도록 수정해보자.
2. [실습8-12]의 스택 크기를 10으로 늘려서 수정해보자.

질문은 이메일을 이용해주세요.
ds.june2@gmail.com

감사합니다