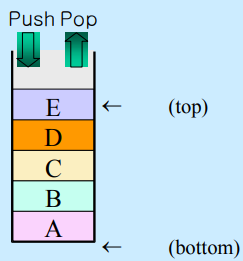
C프로그래밍 실습

**#EX2 Stack**

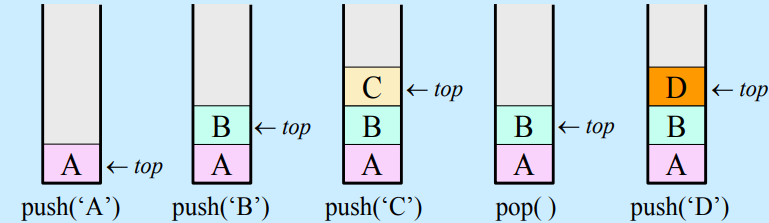
* Stack

데이터에 접근하는 것이 오로지 stack의 마지막(top)만 가능하다.

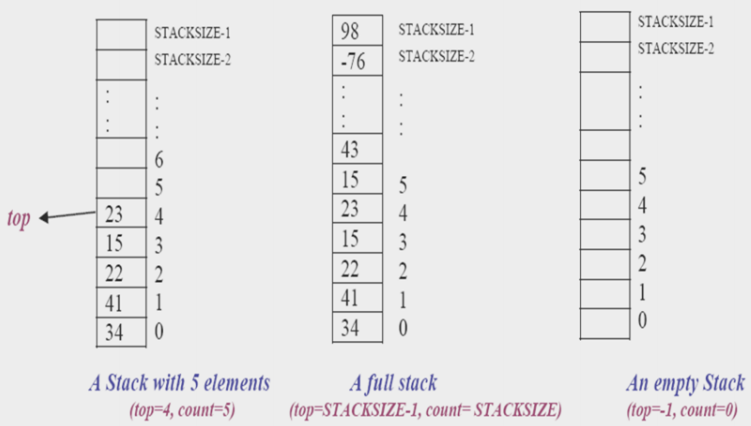
스택의 또다른 마지막은 bottom이다.

+ LIFO(Last-In-First-Out)

* Push(Data), Data=pop()
* An Example of Stack usage



* Stack as an Array



+ stack 배열과 top 변수는 모든 스택 함수에서 접근 가능해야 한다. (전역변수)

+ top변수의 범위

- 프로그램이 시작될 때 (stack이 비었을 때): -1

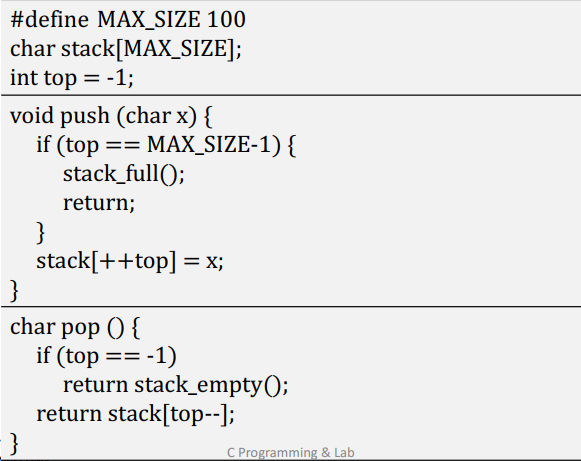
- stack이 찼을 때: [stack 배열 크기 -1]

스택 배열 크기에 대한 매크로 정의를 한다 #define MAX\_SIZE 100

스택이 꽉 찼을 때: top == (MAX\_SIZE-1)

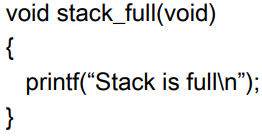
- top변수가 양수, 음수가 될 수 있기 때문에 데이터형을 int로 한다.

* Stack Implementation for char Data



+ void stack\_full(void)

stack이 가득 찼을 때 호출한다: 스택이 가득참을 알려주는 에러메시지를 출력



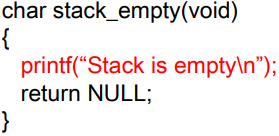
+ char stack\_empty(void)

pop() 연산에서 “top”의 데이터를 return 한다.

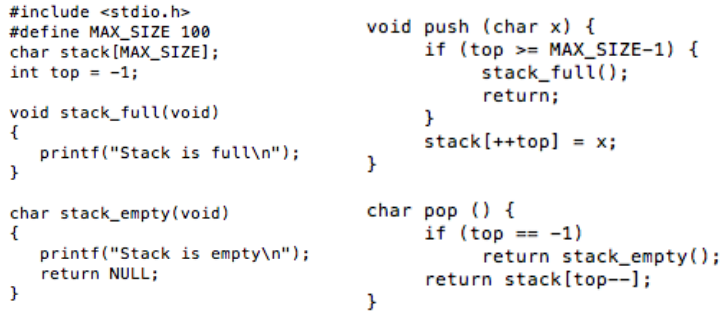
만약 stack이 비었을 때, 호출하면 오류를 반환해야 한다.

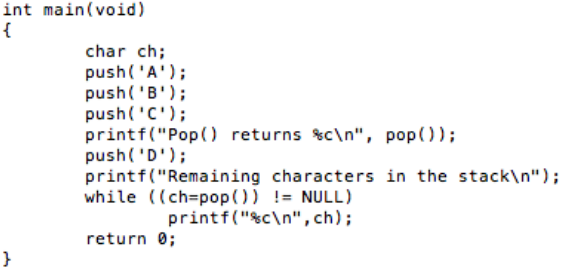
프로그램에서 stack이 빈 상태를 NULL로 가정하자

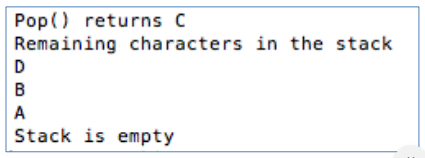
보통 NULL은 NOTHING이라는 의미이다.



* An Example of Stack Programming







* Checking Balanced Parentheses in Expressions Using a Stack

stack을 이용해서 괄호가 균형이 맞는지를 확인 할 수 있다.

+ 프로그램은 왼쪽에서 오른쪽으로 표현식을 읽는다.

만약 (가 읽어진다면 push (

만약 )가 읽어진다면 이전에 push되었던 (를 pop한다.

+ 표현식의 모든 문자를 읽고 스택이 비어있으면 균형이 맞는 괄호이다.

그렇지 않으면 stack이 비어있지 않으니 괄호가 균형이 맞지 않는다.

+ 이미 push 된 것이 pop되었을 때, 그리고 stack이 비었는데 pop되었을 때

스택에 없다면 괄호안의 균형이 맞지 않는다.

