#include <stdio.h>

#define MAX\_STACK\_SIZE 3

typedef int element;

typedef struct {

element stack[MAX\_STACK\_SIZE];

int top;

} StackType;

// 스택 초기화 함수

void init(StackType \*s)

{

s->top = -1;

}

// 공백 상태 검출 함수

int is\_empty(StackType \*s)

{

return (s->top == -1);

}

// 포화 상태 검출 함수

int is\_full(StackType \*s)

{

return (s->top == (MAX\_STACK\_SIZE-1));

}

// 삽입함수

void push(StackType \*s, element item)

{

if( is\_full(s) ) {

fprintf(stderr,"스택 포화 에러\n");

return;

}

else s->stack[++(s->top)] = item;

}

// 삭제함수

element pop(StackType \*s)

{

if( is\_empty(s) ) {

fprintf(stderr, "스택 공백 에러\n");

exit(1);

}

else return s->stack[(s->top)--];

}

element peek(StackType \*s)

{

if( is\_empty(s) ) {

fprintf(stderr, "스택 공백 에러\n");

exit(1);

}

else return s->stack[s->top];

}

**void stack\_print(StackType \*s)**

**{**

**int i;**

**if (is\_empty(s))**

**printf("(empty)\n");**

**else {//스택이 비어있지 않으면 top을 출력하고 나머지를 출력**

**printf("%d <- top\n", s->stack[s->top]);**

**for (i = s->top - 1; i >= 0; i--)**

**printf("%d\n", s->stack[i]);**

**}**

**printf("--\n");**

**}**

// 주함수

void main()

{

StackType s;

init(&s);

stack\_print(&s);

push(&s, 10);

stack\_print(&s);

push(&s, 20);

stack\_print(&s);

push(&s, 30);

stack\_print(&s);

push(&s, 40);

stack\_print(&s);

pop(&s);

stack\_print(&s);

push(&s, 50);

stack\_print(&s);

pop(&s);

stack\_print(&s);

pop(&s);

stack\_print(&s);

pop(&s);

stack\_print(&s);

}