

## 05

## 프로그래밍 프로젝트

**01** 배열로 구현된 스택을 사용하여 다음과 같은 순서로 프로그램을 작성하라.

(1) 다음과 같은 순서로 정수를 스택에 삽입, 삭제하도록 프로그램 5.3의 `main` 함수를 변경하라. 배열의 크기인 `MAX_STACK_SIZE`는 3으로 한다.

10 삽입→20 삽입→30 삽입→40 삽입→삭제→50 삽입→삭제→삭제→삭제

(2) 현재 스택에 들어 있는 요소들을 다음과 같은 형식으로 출력하는 함수 `stack_print(StackType *s)`를 작성하라. 요소를 (1)과 같이 삽입, 삭제할 때마다 `stack_print()` 함수를 호출하여 현재 스택의 모습을 출력하라.

30 <-top

20

10

(3) 스택에 다음과 같이 정수와 문자열을 동시에 저장하려고 하면 앞의 프로그램을 어떻게 변경하여야 하는가? (힌트: 프로그램 5.2처럼 구조체 스택을 사용한다.)

(10, "ten") 삽입 → (20, "twenty") 삽입 → (30, "thirty") 삽입 → (40, "forty") 삽입  
→ 삭제 → (50, "fifty") 삽입 → 삭제 → 삭제 → 삭제

**02** 연결 리스트로 구현된 스택에 대하여 문제 1의 (1)부터 (3)까지를 되풀이하라.

**03** 배열로 구현된 스택과 연결 리스트로 구현된 스택에 대하여 수행시간을 각각 측정하여 보자. 수행시간을 측정하기 위하여 스택에 100,000번 정도의 삽입과 삭제를 되풀이하는 프로그램을 작성하라. 배열로 구현된 스택과 연결 리스트로 구현된 스택에 대하여 수행시간을 기록하라. 어느 것이 더 빠른가? 그리고 그 이유는?

```
time_t start, finish;
double duration;
int i;

init(&s);
start=clock();
for(i=0; i<1000000; i++){
    push(&s, 10);
    pop(&s);
}
finish=clock();
duration = (double)(finish - start) / CLOCKS_PER_SEC;
printf("%f 초입니다.\n", duration);
```

**04** 회문(palindrome)이란 앞뒤 어느 쪽에서 읽어도 같은 말련막문등을 의미한다. 예를 들면 "eye". "madam, I'm Adam", "race car" 등이다. 여기서 물론 구두점이나 스페이스, 대소문자 등은 무시하여야 한다. 스택을 이용하여 주어진 문자열이 회문인지 아닌지를 결정하는 프로그램을 작성하라.

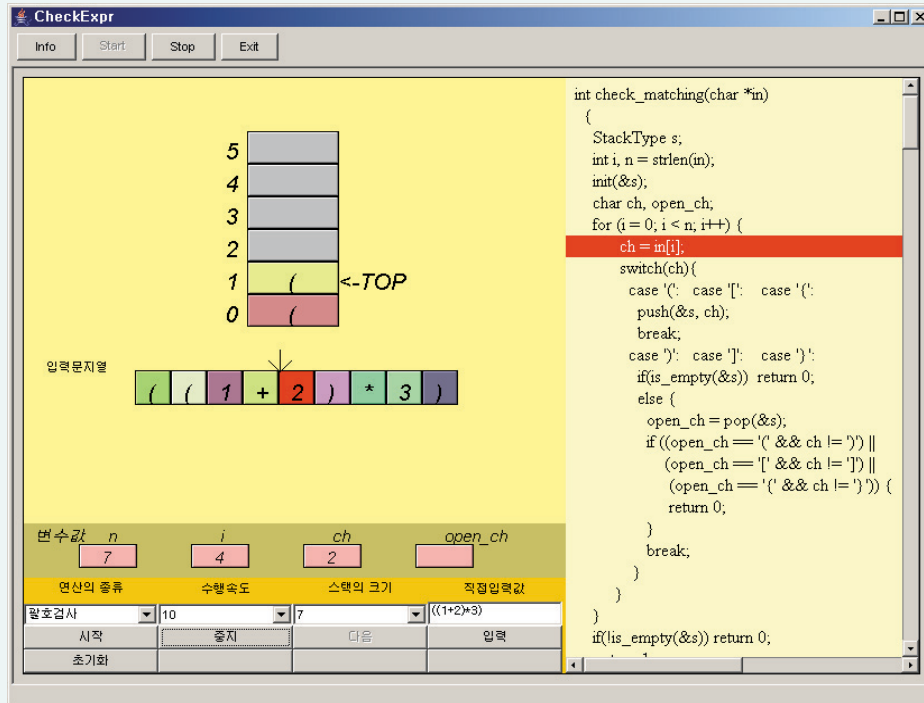
(힌트) 많은 방법이 있을 수 있다. 스택에 삽입했다 꺼내면 순서가 반대로 된다는 것을 이용하여야 한다. 하나의 방법은 입력 문자열의 모든 문자를 스택에 삽입한 다음, 스택에서 문자들을 다시 꺼내면서 입력 문자열의 문자와 하나씩 맞추어보는 것이다. 만약 한 문자라도 맞지 않으면 회문이 아니다. 여기서 주의할 점은 스택에 삽입하기 전에

만약 문자가 스페이스거나 구두점 이면 스택에 삽입하지 말아야한다. 그리고 대소문자를 처리하기 위하여 문자들을 비교하기 전에 모든 문자를 소문자로 변경한다.

```
// 회문이면 TRUE 반환 아니면 FALSE를 반환한다.
int palindrome(char in_str[])
{
    StackType s;
    int i;
    char ch, chs;
    int len=strlen(in_str);

    _____ ;    // 스택을 초기화하라
    for(i=0;i <len; i++){
        ch=in_str[i];
        // 만약 ch가 스페이스거나 구두점이면
        if(_____) continue;
        ch = tolower(ch); // ch를 소문자로 변경
        _____; // 스택에 삽입한다.
    }
    for(i=0;i <len; i++){
        ch=in_str[i];
        // 만약 ch가 스페이스거나 구두점이면
        if(_____) continue;
        ch = tolower(ch); // ch를 소문자로 변경
        chs = _____; // 스택에서 문자를 꺼낸다
        if( ch != chs ) return FALSE; // 실패
    }
    return TRUE; // 성공
}
```

\* CD 부록에 포함된 괄호검사 애플릿을 이용하여 본문에 나오는 괄호 검사 알고리즘을 실험하여 보자.



1. “직접입력” 필드에 검사하고자 하는 수식을 입력한다.
2. 수행 속도를 단계실행이나 적당한 속도로 설정한 다음, “다음” 버튼이나 “시작” 버튼을 눌러서 프로그램을 수행시킨다.
3. 프로그램 코드가 실행되면서 스택의 내용이 어떻게 변화되는지를 살펴본다.