

## Computer Concept and Practice Homework #1

(Due: 2015/06/10 23:59)

### Problem 1. Interactive Calculator

구현하고자 하는 프로그램은 다음과 같이 작동한다.

1. 먼저, 다음과 같이 메시지를 출력하고, 해당되는 operator를 키보드로 입력받는다.

Input operator :

지원하는 operator는 다음의 12가지이다.

+: operand 2개를 입력 받는 연산자. 두 operand를 더한 값을 출력하는 operator이다.

-. operand 2개를 입력 받는 연산자. operand 1에서 operand 2를 뺀 값을 출력하는 operator이다.

\*: operand 2개를 입력 받는 연산자. 두 operand를 곱한 값을 출력하는 operator이다.

/: operand 2개를 입력 받는 연산자. operand 1을 operand 2로 나눈 몫을 출력하는 operator이다.

%. operand 2개를 입력 받는 연산자. operand 1을 operand 2로 나눈 나머지를 출력하는 operator이다.

x: 화면에 다음과 같이 message를 출력하고 program을 끝내는 입력이다.

Program terminates...

이상의 6개의 입력은 program 내에 저장된 값과 무관한 연산자이며, 이하의 6개의 입력은 program이 관리하는 하나의 값을 manipulate하는 연산자이다.

a: operand 1개를 입력 받는 연산자. Program 내에 저장된 값에 해당 operand를 더해서 저장하고 출력한다.

s: operand 1개를 입력 받는 연산자. Program 내에 저장된 값에서 해당 operand를 빼서 저장하고 출력한다.

t: operand 1개를 입력 받는 연산자. Program 내에 저장된 값에 해당 operand를 곱해서 저장하고 출력한다.

d: operand 1개를 입력 받는 연산자. Program 내에 저장된 값을 해당 operand로 나누어서 그 몫

을 저장하고 출력한다.

m: operand 1개를 입력 받는 연산자. Program 내에 저장된 값을 해당 operand로 나누어서 그 나머지를 저장하고 출력한다.

c: 저장된 값을 0으로 clear한 후, 화면에 다음과 같은 메시지를 출력한다.

The value is cleared to 0.

프로그램의 단순화를 위해 operand는 정수만 들어온다고 가정하고, **operator 부분에서는 한 글자의 입력만 들어온다고 가정한다(2015-06-01 15:00 수정).**

입력받는 operator가 영문자일 경우 case-insensitive하게 처리하도록 하며, 위의 12종류 외의 입력이 들어왔을 경우, 에러 메시지를 출력하고 그 다음 줄에 다시 Input operator: 를 출력하여 입력받는다.

Program 이 관리하는 하나의 값의 초기값(위의 어떤 연산도 가해지지 않았을 때의 value)은 0으로 한다.

2. Operator를 입력받은 후, 다음과 같이 출력하여 operand를 한 개 또는 두 개를 키보드로 입력받는다.

Operand를 1개 입력받을 때에는 다음과 같이 출력한다.

Operand 1:

Operand를 2개 입력받을 때에는 다음과 같이 출력한다.

Operand 1,2:

Operand를 1개 입력받을 때에는 정수 1개를 입력받는다.

operand를 2개 입력받을 때에는 정수 2개를 콤마(,)를 구분자로 하여 입력받는다.

이 때 0으로 나누는 입력은 들어오지 않는다고 가정한다.

**이 부분에 대해서는 올바른 입력만 들어온다고 가정한다. (2015-06-01 15:00 수정)**

3. 입력받은 operator와 operand(s)를 가지고 1번에서 제시된 동작을 수행하여 결과를 화면에 출력한다.

4. 프로그램을 계속 작동시킬 것인가를 묻는 메시지를 출력한 후, 그 여부를 입력받는다.

```
Continue? (y/n)
```

이 때, y 또는 n만을 case-sensitively 입력받도록 하며, 그 외의 입력이 들어왔을 경우에는 에러 메시지를 출력하고 그 다음 줄에 다시 Continue? (y/n) 을 출력하여, 계속 프로그램을 작동시킬 것인지에 대한 여부를 묻도록 한다. y를 입력받으면 1번으로 돌아가 새로운 operator를 입력받는 것을 시작으로 작동을 계속 하면 되며, n를 입력받으면 다음과 같이 메시지를 출력하고 프로그램을 종료하도록 한다.

```
Program terminates...
```

example 작동은 다음과 같다.

```
Input operator: +
Operand 1,2: 3,5
8
Continue? (y/n) y
Input operator: a
Operand 1: 3
3
Continue? (y/n) y
Input operator: a
Operand 1: 5
8
Continue? (y/n) y
Input operator: c
The value is cleared to 0.
Continue? (y/n) Y
Error: Continue input wants y or n.
Continue? (y/n) y
Input operator: t
Operand 1: 10000
0
Continue? (y/n) y
Input operator: +
```

```
Operand 1,2: 10,20
30
Continue? (y/n) y
Input operator: -
Operand 1,2: 30,-10
40
Continue? (y/n) n
Program terminates...
```

## Problem 2. Interactive Calendar

구현하고자 하는 프로그램은 다음과 같이 작동한다.

1. 먼저, 다음과 같이 메시지를 출력하고, 해당되는 year와 month를 슬래시(/)를 구분자로 하여 키보드로 입력받는다.

```
Input year/month:
```

이 때 year는 자연수로, month는 1부터 12까지의 자연수로 가정하며, 그 외의 값이 들어왔을 경우 에러 메시지를 출력하고 다시 Input year/month: 를 출력하여 해당 값을 입력받도록 한다.

(참고: 서기 1년 1월 1일은 월요일이다.)

2. 해당되는 년월의 달력을 출력한다.

이 때 달력은 보기 편리하게 출력한다. 올바른 달력임을 알아볼 수 있으면 된다.

3. 프로그램을 계속 작동시킬 것인가를 묻는 메시지를 출력한 후, 그 여부를 입력받는다.

```
Continue? (y/n)
```

이 때, y 또는 n만을 case-sensitively 입력받도록 하며, 그 외의 입력이 들어왔을 경우에는 에러 메시지를 출력하고 그 다음 줄에 다시 Continue? (y/n) 을 출력하여, 계속 프로그램을 작동시킬 것인지에 대한 여부를 묻도록 한다. y를 입력받으면 1번으로 돌아가 새로운 operator를 입력받는 것

을 시작으로 작동을 계속 하면 되며, n를 입력받으면 다음과 같이 메시지를 출력하고 프로그램을 종료하도록 한다.

```
Program terminates...
```

example 작동은 다음과 같다.

```
Input year/month: 2015/5
      May 2015
-----
S   M   T   W   T   F   S
                1   2
 3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
31
-----
Continue? (y/n) y
Input year/month: 2015/4
      April 2015
-----
S   M   T   W   T   F   S
                1   2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30
-----
Continue? (y/n) n
Program terminates...
```

(Problem 1,2 공통 주의사항)

에러 메시지는 어떤 에러가 발생했는지 알 수 있는 내용을 포함하도록 한다. 이는 곧 어떤 에러가 발생했는가에 따른 에러 메시지가 나타나야 함을 의미한다.

채점시 프로그램의 올바른 작동 여부를 최우선으로 하므로 최대한 correctness를 만족시킬 수 있도록 작성하는 것이 중요하다.

컴파일에서 에러가 나면 무조건 0점이 부여되므로 컴파일에서 에러가 나지 않도록 하며, 경고 (warning)도 최대한 나타나지 않도록 한다.

#### 제출 방법

code 파일의 이름은 1번 문제는 <학번>\_prob1.c, 2번 문제는 <학번>\_prob2.c 로 한다.

code 파일들을 HW1\_<학번>.zip으로 압축 후 첨부하여 [hglee@archi.snu.ac.kr](mailto:hglee@archi.snu.ac.kr)로 제출하도록 한다.

메일 제목은 [검개실]HW1\_<학번>\_<이름> 으로 한다.

위의 <학번>자리에 자신의 학번을, <이름>자리에 자신의 이름을 넣으면 된다.

제출이 Due보다 24시간 늦어질 때마다 다음과 같은 방식으로 10점씩 감점한다.

Due+1초~24시간 : 10점 감점

Due+24시간1초~48시간 : 20점 감점

...

#### 채점 환경

채점 환경은 제공된 서버로 한다.