Computer Concept and Practice Homework #1

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

Problem 1. Interactive Calculator

The program you implement should operate following features.

1. First, you output this message and receive an operator from keyboard.

```
Input operator :
```

Supported operators are following 12 operators:

+: for 2 operands. The result is the value that added received two values.

-: for 2 operands. The result is the value that subtracted operand 2 from operand 1.

*: for 2 operands. The result is the value that calculated by operand 1 times operand 2.

/: for 2 operands. The result is the quotient that calculated by dividing operand 1 with operand 2.

%: for 2 operands. The result is the remainder that calculated by dividing operand 1 with operand 2.

x: terminating input that outputs this message:

```
Program terminates…
```

Above six operators are not relevant with a stored value in your program, and following six operators are operators that manipulate the stored value managed by your program.

a: for 1 operand. You add the operand to your stored value, change the value to the new value, and output it.

s: for 1 operand. You subtract the operand from your stored value, change the value to the new value, and output it.

t: for 1 operand. You multiply the operand to your stored value, change the value to the new value, and output it.

d: for 1 operand. You change your stored value to the quotient calculated by dividing the stored value with the operand and output it.

m: for 1 operand. You change your stored value to the remainder calculated by dividing the stored value with the operand and output it.

c: Clear your stored value to 0 and output following message:

```
The value is cleared to 0.
```

For simplification, assume that operands are integers, **and the input for operator is only one letter(modified at 15:00, June 1st, 2015)**.

The input operator is case-insensitive, and if the input for operator is different with 12 kinds of inputs, print error message and from the next line repeat from the first.

The initial value of stored value in your program is 0.

2. After receiving an operator, print following and receive operand(s).

When receiving operand is one, print following:

```
Operand 1:
```

When receiving operands are two, print following:

```
Operand 1,2:
```

When receiving operands are two, separator between them is a comma.

Assume that there is no 0-division input.

Assume that all inputs are correct. (modified at 15:00, June 1st, 2015)

3. Print results.

4. Print asking message like following which asks to continue the program or not.

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

Receive y or n case-sensitively, and if received input is other than y or n, print error message and re-ask. When input is y, repeat from 1., and when input is n, print following message and terminate your program.

```
Program terminates...
```

Example program operation:

```
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

The program you implement should operate following features.

1. First, print following message and receive year and month with a separator, slash(/), from keyboard.

```
Input year/month:
```

The input for year is a natural number, and month is between 1 and 12. When other inputs are received, print error messages and repeat from the first.

(Hint: January 1st, 1 is Monday.)

2. Print calendar of received month of received year.

Your printed calendar should be recognizable and correct.

3. Print asking message like following which asks to continue the program or not.

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

Receive y or n case-sensitively, and if received input is other than y or n, print error message and re-ask. When input is y, repeat from 1., and when input is n, print following message and terminate your program.

```
Program terminates...
```

Example program operation:

```
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...
```

(Caution)

An error message should notify that which error is occurred.

When grading, program's correctness has first priority, so work hard to satisfy program's correctness.

When there is any error in your submitted program's compilation, your program will be graded to the score 0.

Avoid any warnings in your program's compilation.

Submission

Names of code files: <student_id>_prob1.c, <student_id>_prob2.c

Compress your codes with the file <student_id>.zip and submit to hglee@archi.snu.ac.kr.

Mail title should be [CCP]HW1_<student_id>_<your_name>.

Example: 2015-12345_prob1.c, 2015-12345_prob2.c are compressed to 2015-12345.zip

Mail title example: [CCP]HW1_2015-12345_Alice

When your submission is late, your homework score will be degraded by 10 points according to your submission time.

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

Due+1second~24hours : -10 points from your homework score

Due+24hours 1second~48hours : -20 points from your homework score

...

Grading Environments

Grading will be done in the provided server.