

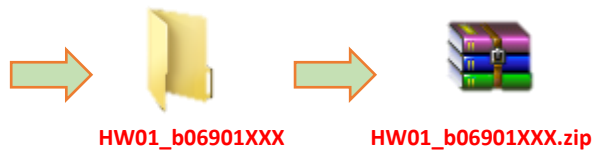
Computer Programming Homework #1

2017/10/25 by TA 陳泓弦 陳姿玲

REQUIRED FILES

Please **compress a folder** named **HW01_b06901XXX** (student ID) that contains the following files:

- ✓ b06901XXX_p1.cpp
- ✓ b06901XXX_p2.cpp
- ✓ b06901XXX_p3.cpp
- ✓ b06901XXX_p4.cpp
- ✓ b06901XXX_p5.cpp
- ✓ b06901XXX_p6.cpp



Do not submit executable files (.exe). Files with names in the wrong format will not be graded. In your .cpp files, we suggest you write comments in details as much as you can. It will be good for TAs to read your code and for your future reference and maintenance.

Homework #1 is due on November 12 (Sunday).

PROBLEM DESCRIPTION

Problem 1. [15 points] [Require: b06901XXX_p1.cpp]

Please write a program that converts the decimal number into a base-N system. Note N is in the range of 1 to 16 and is an integer. The program should read two integers such that one is the decimal number and the other is the base system N which you would like to convert to. For base 16, A denotes 10, B denotes 11 and so on.

Format:

```
Please input a positive decimal number: 264
Please input the desired base :3
"264" in base-10 system = "100210" in base-3system
```

Problem 2. [20 points] [Require: b06901XXX _p2.cpp]

Today is the last day of the summer vacation. However, Jeffery totally forgot doing his summer math homework! There are 1000 exercises about prime factorization (質因數分解), greatest common divisor (gcd, 最大公因數) and least common multiple (lcm, 最小公倍數). It is impossible to finish all of them by handwriting on one night. Luckily, Jeffery had learned C++ and he could write C++ programs to do computation instead hand calculation. Please write a program that performs the following two tasks:

- (a) The first task is to calculate the “prime factorization”. The input is a positive integer and you have to print out the result as the following format.

Format:

```
(a)
Please input a positive integer: 2700
2700 = 2^2 x 3^3 x 5^2
```

- (b) The second task is to calculate the “greatest common divisor” (gcd) and “Least Common Multiple” (lcm). The inputs are two positive integers and you have to print out the results of gcd and lcm respectively as the following format.

Format:

```
(b)
Please input two positive integers: 120 78
gcd (120,78) = 6
lcm (120,78) = 1560
```

Note: Please write (a) (b) in one program in the same .cpp file. The output format should look like the following.

```
(a)
Please input a positive integer: 2700
2700 = 2^2 x 3^3 x 5^2
(b)
Please input two positive integers: 120 78
gcd (120,78) = 6
lcm (120,78) = 1560
```

Problem 3. [15 points] [Require: b06901XXX_p3.cpp]

You must have used the calculator before and know that the numbers displayed on the calculator are in a special letter format as follows. Please write a program to print the number as the calculator would display. The input is a positive integer with the number digits less than 10.

Format:

```
Please input a positive integer (digit<10):456789
  --  --  --  --  --
|  |  |  |  |  |  |  |
--  --  --  |  --  --
  |  |  |  |  |  |  |
  --  --  --  --  --
```

Letter format:

```
  --  --  --  --  --  --  --  --  --
|  |  |  |  |  |  |  |  |  |
|  |  |  --  --  --  --  --  |  --  --
|  |  |  |  |  |  |  |  |  |
--  --  --  --  --  --  --  --  --
```

Problem 4. [15 points] [Require: b06901XXX_p4.cpp]

Please write a program to count the occurrence of English alphabets in the input sentence. The input sentence has less than 100 characters and different words are separated by using the '/' symbol. (special symbol like '?', '!', '\' is counted as one character). The upper-case and lower-case alphabets are considered as same alphabet (i.e. 'A' and 'a' is same). For example, if the input paragraph is "**How/are/you?**", each of the alphabets A, E, H, R, U, W, Y appears one time and O appears two times. Other characters such as '?', '!', '+' ... will be neglected. After reading the sentence, your program should print out the occurrence of each alphabet and the number of different alphabets in your input sentence. You should print alphabets in the order of A to Z and the output changes to next line every 3 alphabets as the following format. **You can only use cin to get the input sentence.** Other functions for reading input will get 0 point. Space will not appear in TA's testing data.

Format:**example 1**

```
How/are/you?
A's count = 1   E's count = 1   H's count = 1
O's count = 2   R's count = 1   U's count = 1
W's count = 1   Y's count = 1
There are 8 different alphabets in your input.
```

example 2

```
Yeah!/I/have/finished/my/C++/homework.
A's count = 2   C's count = 1   D's count = 1
E's count = 4   F's count = 1   H's count = 4
I's count = 3   K's count = 1   M's count = 2
N's count = 1   O's count = 2   R's count = 1
S's count = 1   V's count = 1   W's count = 1
Y's count = 2
There are 16 different alphabets in your input.
```

Problem 5. [20 points] [Require: b06901XXX_p5.cpp]

In the well lived two snails (A & B). One day two snails suddenly want to play a game about “Who can climb out the well first”. The well is very deep, so snails need to climb up the well in the morning and take rest in the evening in a day. The next day when snails start climbing, they find that they slip down for a fixed distance because of the gravitational effect. In addition, snails would feel tired for climbing so the climbing distance would be decreased in every morning. For example, the snail climbs up 3 meters in the morning, slip down 2 meters in the evening, and the climbing distance is decreased 20 cm on the next day as the following table.

Day	The height to start	After climbing up	After slipping down
1	0 m	3 m	1 m
2	1 m	3.8 m	1.8 m
3	1.8 m	4.4 m	2.4 m
4	2.4 m	4.8 m	2.8 m

Write a program to find out the final status of the two snails as the following format.

Format:

```
The well's height(meter):XX.xx <--(meter)
Snail A's information:
XX.xx <--(climbing up distant of A)
XX.xx <--(slipping down distant of A)
XX.xx <--(decreased distant due to feeling tired of A)
Snail B's information:
XX.xx <--(climbing up distant of B)
XX.xx <--(slipping down distant of B)
XX.xx <--(decreased distant due to feeling tired of B)
On day ## : Snail A "climb up to the top" or "slip down to the bottom"
On day ## : Snail B "climb up to the top" or "slip down to the bottom"
```

Example:

```
The well's height(meter):4.3
Snail A's information:
3.0
2.0
0.2
Snail B's information:
2.0
0.7
0.3
On day 3:Snail A climb up to the top
On day 10:Snail B slip down to the bottom
```

Problem 6. [15 points] [Require: b06901XXX_p6.cpp]

Please write a little program to find the treasure! First you have to build a terrain as in map.fig. The terrain is a 5*5 space and 1, 2, 3, 4, 5 is for the coordinates of x and y . The symbol of '#' is the boundary, 'X' is your position and '*' is the position of the treasure. At the beginning, you are at the position of (0,0) and the position of the treasure is (5,5). Every time you input two integers, where the first integer is the number of steps in the direction of x and second integer is the number of steps in the direction of y . Your program should take care of the condition when you are about to go outside the boundary.

map.fig:

```
#####
# 5 - - - - * #
# 4 - - - - - #
# 3 - - - - - #
# 2 - - - - - #
# 1 - - - - - #
# X 1 2 3 4 5 #
#####
```

Example:

```
Try to find the treasure! * is the position of the treasure.
You are at the origin. X is your position.
#####
# 5 - - - - * #
# 4 - - - - - #
# 3 - - - - - #
# 2 - - - - - #
# 1 - - - - - #
# X 1 2 3 4 5 #
#####
Move on direction of x, y = 3 1
    |
    |
    V
#####
# 5 - - - - * #
# 4 - - - - - #
# 3 - - - - - #
# 2 - - - - - #
# 1 - - X - - #
# 0 1 2 3 4 5 #
#####
Your position is (3,1)
Move on direction of x, y = 2 2
    |
    |
    V
```

```
#####
# 5 - - - - * #
# 4 - - - - - #
# 3 - - - - X #
# 2 - - - - - #
# 1 - - - - - #
# 0 1 2 3 4 5 #
#####
Your position is (5,3)
Move on direction of x, y = 2 0
    |
    |
    V
#####
# 5 - - - - * #
# 4 - - - - - #
# 3 - - - - X #
# 2 - - - - - #
# 1 - - - - - #
# 0 1 2 3 4 5 #
#####
Your position is (5,3)
Move on direction of x, y = 0 2
    |
    |
    V
```

```
#####
# 5 - - - - X #
# 4 - - - - - #
# 3 - - - - - #
# 2 - - - - - #
# 1 - - - - - #
# 0 1 2 3 4 5 #
#####
Your position is (5,5)
SUCCESSFUL!
```