1st Mid-term Exam

C COMPUTER PROGRAMMING LABORATORY （I）

October 26, 2023

Exam rules

● Only Dev-C++ can be used for the exam.
● If your codes cannot be compiled by Dev-C++, it is considered as syntax error.
● Please write all your codes as a c source file named after your student ID.
  example: M113040076_1.c、M113040076_2.c…
● When submitting the assignment, upload all seven questions together to the university's assignment submission area, without the need to compress them.
● **In a single question, the score is either full credit or zero.**
● Please check whether your codes can be compiled and output the desired result before submitting to National Sun Yat-sen Cyber University.
● No reason for late submission.
● It is forbidden to search for information during the exam.
● **The cheaters will get zero point.**

---

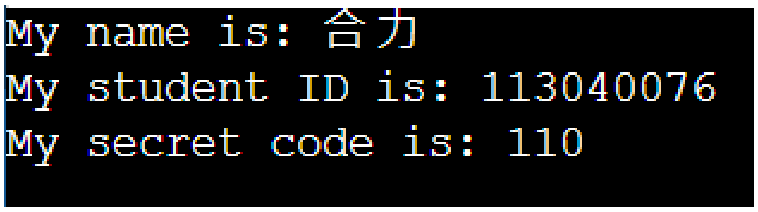
a. (10 points)

Description

Design a program to print your name and student ID (excluding the first letter). Then print your **secret code**.

Notice

Secret code: Sum up all digits in your student ID, then multiply the sum by 5.



```
My name is: 合力
My student ID is: 113040076
My secret code is: 110
```

b. (10 points)

Description

Please enter an integer N and print the corresponding hourglass figure.

Requirement

Please use continuous input.

Output format should follow the example output.

Terminate the program when the input is 0.

Notice

The hourglass is printed only when N is a positive odd number.

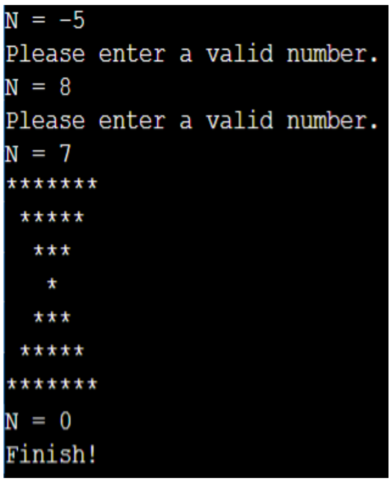
On the other hand, when N is an even number or a negative number, "Please enter a valid number" is printed.

Spaces only need to be printed before *.

Hint

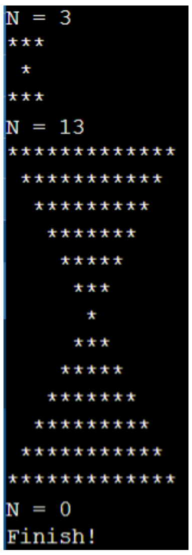
Please use **while(scanf(...) != EOF)**

example1:



```
N = -5
Please enter a valid number.
N = 8
Please enter a valid number.
N = 7
*******
 *****
  ***
   *
  ***
 *****
*******
N = 0
Finish!
```

example2:



```
N = 3
***
 *
***
N = 13
*************
 ***********
  *********
   *******
    *****
     ***
      *
     ***
    *****
   *******
  *********
 ***********
*************
N = 0
Finish!
```

c. (20 points)

Description

For each pair of positive integers N and M input, output their **least common multiple**

Requirement

Please use continuous input.

Output format should follow the example output, and the order of the output should be correct.

Terminate the program when the input is 0 0.

Notice

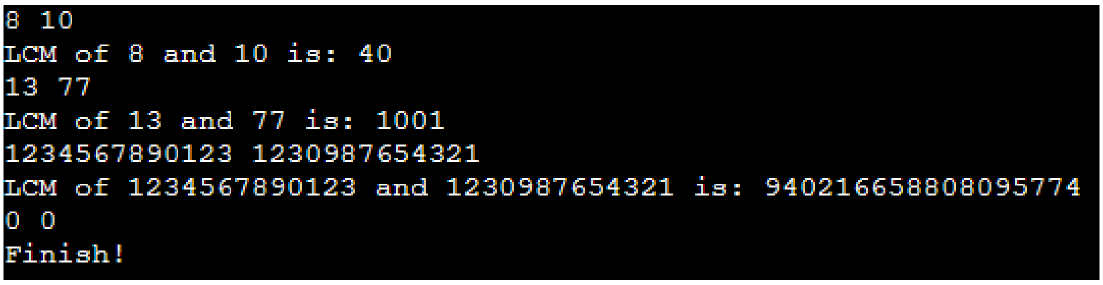
The end of the line has a newline character '\n'.

There will be no empty test cases and invalid test cases.

Each test case consists of two positive integers, both greater than 0 and less than 10,000,000,000,000.

Hint

Please use **while(scanf(...) != EOF)**



```
8 10
LCM of 8 and 10 is: 40
13 77
LCM of 13 and 77 is: 1001
1234567890123 1230987654321
LCM of 1234567890123 and 1230987654321 is: 940216658808095774
0 0
Finish!
```

d. (20 points)

Description

Please write a C program to calculate the area of the **trapezoid** enclosed by the coordinates of four points.

Requirement

Please use continuous input.

Output format should follow the example output, and the order of the output should be correct.

Terminate the program when the input is 0 0 0 0 0 0 0 0.

Notice

There will be no oblique trapezoid (the upper and lower bottoms are parallel to the x-axis or y-axis).

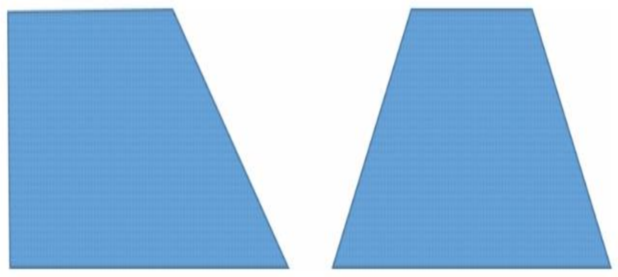
The four coordinate points have no fixed order.

Trapezoidal area formula: (upper base + lower base) * height / 2.

Round the area to two decimal places.

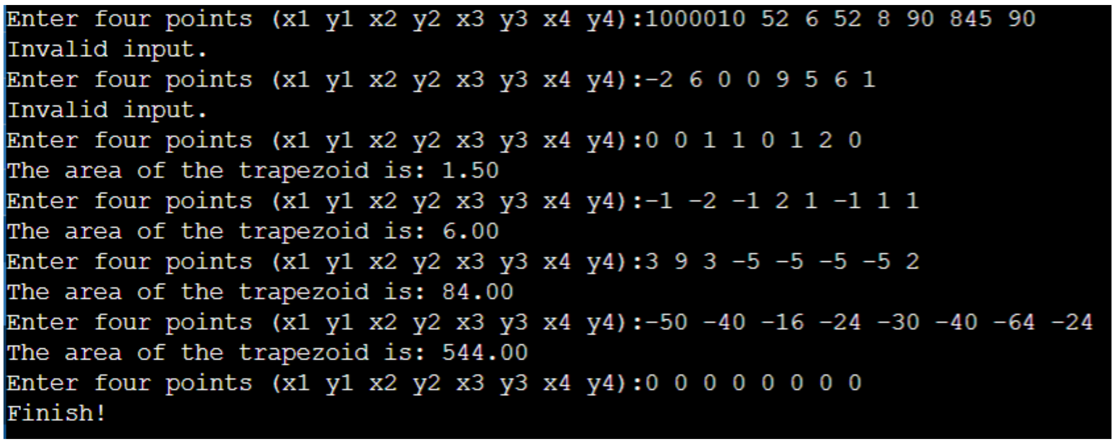
For each x, y must follow the range: -1,000,000 <= x, y <= 1,000,000

There are only the following two types of trapezoids, but they may be upside down, left and right, rotated 90, 180 degrees, etc.



Hint

Please use **while(scanf(...) != EOF)**



```
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):1000010 52 6 52 8 90 845 90
Invalid input.
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):-2 6 0 0 9 5 6 1
Invalid input.
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):0 0 1 1 0 1 2 0
The area of the trapezoid is: 1.50
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):-1 -2 -1 2 1 -1 1 1
The area of the trapezoid is: 6.00
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):3 9 3 -5 -5 -5 -5 2
The area of the trapezoid is: 84.00
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):-50 -40 -16 -24 -30 -40 -64 -24
The area of the trapezoid is: 544.00
Enter four points (x1 y1 x2 y2 x3 y3 x4 y4):0 0 0 0 0 0 0 0
Finish!
```

e. (20 points)

Description

Input two positive integers x and y, and print the **prime numbers** between x and y.

Requirement

Please use continuous input.

The output is sorted in order of small to large , and each prime number in one line

Notice

a) x can't be bigger than y.

b) Neither x nor y can be negative. If you encounter the above situation, you need to output the “Invalid input\n”.

c) The maximum number does not exceed the upper limit of int.

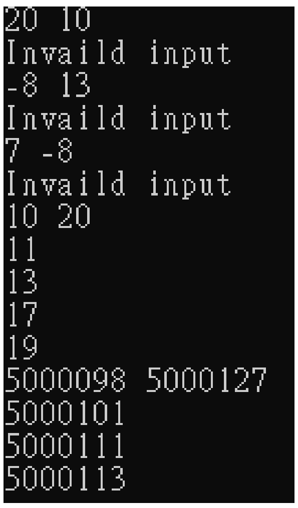
d) If there are no prime numbers between the two numbers, there is no need to output.

e) Line break '\n' at the end of each line.

Hint

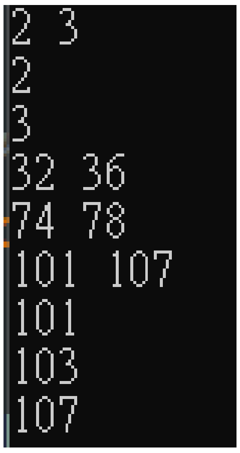
Please use **while(scanf(...) != EOF)**

example 1:



```
20 10
Invaild input
-8 13
Invaild input
7 -8
Invaild input
10 20
11
13
17
19
5000098 5000127
5000101
5000111
5000113
```

example 2:



```
2 3
2
3
32 36
74 78
101 107
101
103
107
```

f. (20 points)

Description

You are climbing a staircase. It takes n steps to reach the top

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?.

Requirement

Please use continuous input.

Output format should follow the example output, and the order of the output should be correct.

Terminate the program when the input is 0.

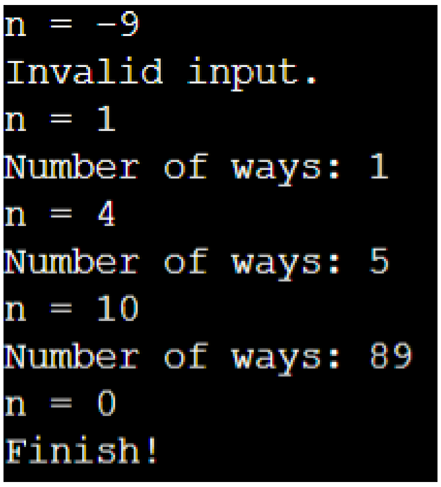
Notice

For each n must follow the range: 0 < n < 45

Output the number of ways to get the top.

Hint

Please use **while(scanf(...) != EOF)**



```
n = -9
Invalid input.
n = 1
Number of ways: 1
n = 4
Number of ways: 5
n = 10
Number of ways: 89
n = 0
Finish!
```

Example 1:

```
Input: n = 2

Output: 2

Explanation: There are two ways to climb to the top.

1. 1 step + 1 step

2. 2 steps
```

Example 2:

```
Input: n = 4

Output: 5

Explanation: There are four ways to climb to the top.

1. 1 step + 1 step + 1 step + 1 step

2. 1 step + 1 step + 2 steps

3. 1 step + 2 steps + 1 step

4. 2 steps + 1 step + 1 step

5. 2 steps + 2 steps
```

g. (20 points)

Description

Input integers and output the results of arithmetic operations based on the program's state.

Requirement

Please use continuous input until EOF.

All inputs are integers, there will be no invalid input.

Notice

Rules:

(State RST) On the 1st input, record the value.

(State ADD) On the 2nd input, output the addition result with the 1st input.

(State SUB) On the 3rd input, output the subtraction result with the 2nd output.

(State MUL) On the 4th input, output the multiplication result with the 3rd output.

(State DIV) On the 5th input, output the division result with the 4th output (5th / 4th). **If the 4th output is 0, only output "division by zero => reset," and proceed to the next step in (State RST).**

(State ADD) On the 6th input, output the addition result with the 5th output.

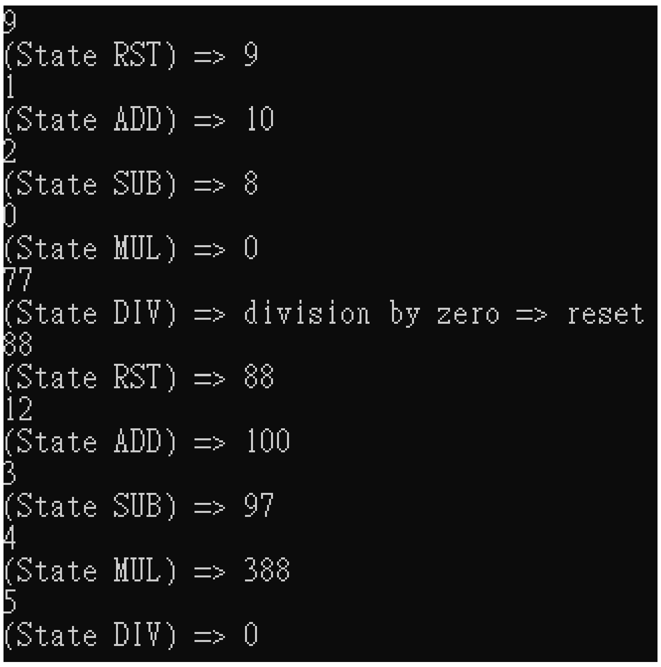
And so on...

Each line of output (current state) => result. E.g. (State N) => RESULT.

Hint

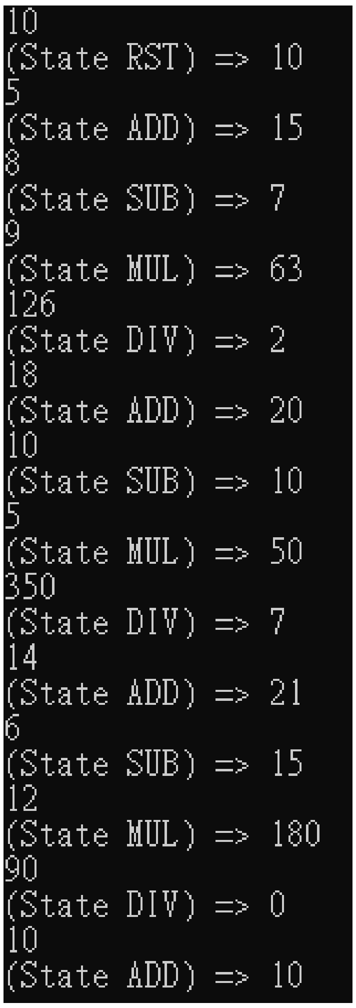
Please use **while(scanf(...) != EOF)**

example 1:



```
9
(State RST) => 9
1
(State ADD) => 10
2
(State SUB) => 8
0
(State MUL) => 0
77
(State DIV) => division by zero => reset
88
(State RST) => 88
12
(State ADD) => 100
3
(State SUB) => 97
4
(State MUL) => 388
5
(State DIV) => 0
```

example 2:



```
10
(State RST) => 10
5
(State ADD) => 15
8
(State SUB) => 7
9
(State MUL) => 63
126
(State DIV) => 2
18
(State ADD) => 20
10
(State SUB) => 10
5
(State MUL) => 50
350
(State DIV) => 7
14
(State ADD) => 21
6
(State SUB) => 15
12
(State MUL) => 180
90
(State DIV) => 0
10
(State ADD) => 10
```