CIS

COMPUTER AND INFORMATION SCIENCES

UAB CIS High School Programming Contest

March 18, 2017

OFFICIAL 10th GRADE AND BELOW CONTEST QUESTIONS

Each problem in this packet contains a brief description, followed by three example test cases of a successful implementation.

The example input and output shown for each question should be regarded only as examples.We will test your programs on the examples provided with each problem, as well as several other test cases generated by the judges.

Be sure to follow the input and output formatting exactly, printing each output answer on its own line.Extraneous or malformed output will result in a failed submission.

If your program fails, a result will be returned by the submission system stating the counterexample test case that caused your program to be judged incorrect.

Each incorrect answer will be incur a 20 minute time penalty, but this penalty will only be applied to your score if the problem is eventually answered correctly.

As noted in the rules, the overall ranks will be determined first by the number of problems completed, and then by your time score (including penalties).

Please pay close attention to the directions in each problem description.

In some cases, assumptions are stated about the limitations of the input, which are designed so that you do not have to consider difficult cases or perform input validation.

If you get stuck on a problem, you are encouraged to jump around and try different problems, as an eventual correct answer on a problem is better than no answer at all.

1. Rectangle Properties

Write a program to take length and width of a rectangle as input from the user.
Calculate and print area, perimeter, and length of diagonal for this rectangle.
Note length and width should be positive.
You may find this equation useful:

$$\text{diagonal}^2 = \text{length}^2 + \text{width}^2$$

The input is given as 2 lines, first line is length and second is width

Sample Input	Sample Output

4
3
	Area = 12
	Perimeter = 14
	Diagonal = 5

2. Leap Year

Write a program to take year as input from the user and print whether this year is a leap year or not. All years which are perfectly divisible by 4 are leap years except for century years (years ending with 00) which is leap year only it is perfectly divisible by 400.

The input is given as 1 line representing the year

Sample Input	Sample Output
2017	NO

3. Fibonacci Series

The Fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …

$F_0 = 0$

$F_1 = 1$

$F_2 = 1 = F_1 + F_0 = 1 + 0 = 1$

$F_n = F_{n-1} + F_{n-2}$

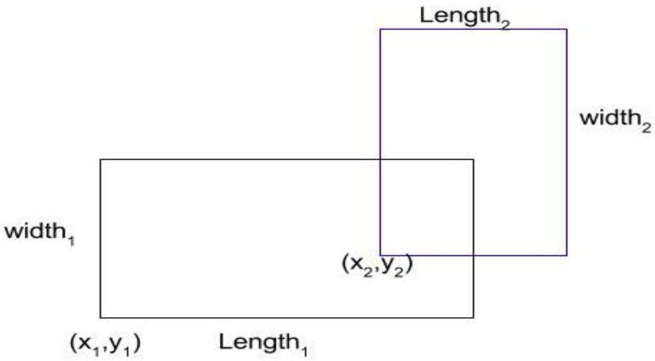
Write a program to take a positive number(n) as input from the user. Print the nth number of the fibonacci series.

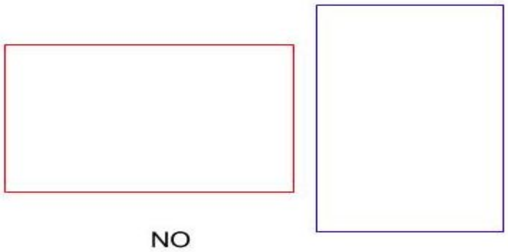
The input is given as 1 line number denoting the nth position

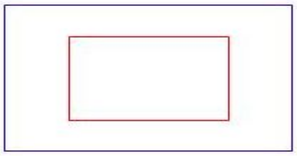
Sample Input	Sample Output
3	2
10	55

4. Rectangles' Overlap

Given two rectangles, determine if they overlap or not.
Write a program where user will input lower left coordinate, width and length for two rectangles. Determine and print YES if they overlap, NO otherwise.
The input is given as 2 lines. First line has 4 space separated integers x, y, length, and width for first rectangle. Second line has 4 space separated integers x, y, length, and width for second rectangle.




NO


YES

Sample Input	Sample Output
10 10 10 5 18 12 8 4	YES

5. Minimum Coins

You are given infinite supply of coins of value (pennies) 1, (nickels) 5, (dimes) 10, and (quarters) 25. You are also given a number in cents which you have to arrive at. Write a program where the user will input this number and your program find and print the minimum number of coins needed to arrive at it.
The input is given as 1 line representing total value in cents.

Sample Input	Sample Output
6	2

6. Intersection and Union of Two Sorted Arrays

Given two sorted integer arrays in ascending order, find their union and intersection. The result must be sorted array too.

Union is set of all unique elements in both arrays (i.e. no duplicates).

Intersection is set of elements found in both arrays.

The input is given as 4 lines. The first line represents the length of the first array, the second line is the array itself, the third line contains the length of the second array, the fourth line contains the second array.

Sample Input	Sample Output

5
1 2 3 4 5
5
1 3 8 9 10
	Union: 1 2 3 4 5 8 9 10
	Intersection: 1 3