UAB CIS High School Programming Contest

March 18, 2017

OFFICIAL HIGH SCHOOL 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 Pairwise Product

Given a target number and an array of unique, positive integers, determine how many pairs exist within the array that can be multiplied to form the target number. For example, if the input array is [2, 4, 1, 35, 8], and the target number is 8, then there exist 2 pairs in the array that can be multiplied together such that their product is 8: [2, 4] and [1, 8]. Thus, the output would be 2.

The input is given as three lines, with the target number being a number on the first line, the size of the array being the number on the second line, and the array being a space-separated list on the second line. Input and output processing has been handled in the sample code, if you wish to use it.

Sample Input	Sample Output
8 5 2 4 1 35 8	2

2 String Shorthand

Given a string containing lowercase letters, convert it to a string containing numbers and letters such that a number, corresponding to the count of a letter, precedes each letter contained in the string. For example, if the input is xxxbbx, the the output should be 3x2b1x because there are 3 x's, followed by 2 b's, followed by 1 x, in the input string.

The input is given as one line containing the input string. Input and output processing has been handled in the sample code, if you wish to use it.

Sample Input	Sample Output
xxxbbx	3x2b1x

3   Fix 56

Given an array that contains positive integers, return an array of the same size as the given array, containing the exact numbers as the input array, but with the numbers rearranged such that every 6 is immediately preceded by a 5. Do not change the position of the 5's in the array, but every other number can be moved. For example, if the input array is [3, 5, 3, 6], then the output array should be [3, 5, 6, 3].

The input is given as a two lines, with the first line containing the size of the array and the second line containing a space-separated list of numbers, corresponding to the input array. The array will contain the same number of 5's and 6's, and there will never be a 5 preceded by another 5. The first 5 will appear in the array before the first 4. The output should be given as a single line containing a space-separated list of numbers, corresponding to the output array. Input and output processing has been handled in the sample code, if you wish to use it.
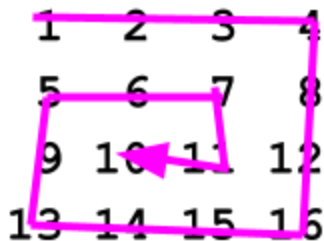
Sample Input	Sample Output
4 3 5 3 6	3 5 6 3

4   Spiral Matrix Traversal

Given a matrix (two-dimensional array) of positive integers, traverse the elements in a clockwise-spiral fashion and output the elements in clockwise-spiral order. For example, if this is the input matrix:

```
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 14 15 16
```

Then the matrix should be traversed as such:

And the output should be this: `1  2  3  4  8  12  16  15  14  13  9  5  6  7  11  10`

The first line of the input contains a number representing the number of rows in the matrix, the second line of the input contains a number representing the number of columns in the matrix, and there are several lines afterward that contain a space-separated list of positive integers corresponding to the rows of the input matrix. The output should be a space-separated list of positive integers, corresponding the the elements of the matrix when it is spirally traversed. Input and output processing has been handled in the sample code, if you wish to use it.

Sample Input	Sample Output
4 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

5   Rainwater Accumulation

The mayor of a two-dimensional city, which is about to have an influx of development in a new area, needs your help. Two-dimensional cities have a problem with rainwater: it can accumulate between buildings, and too much of it will cause the structures to collapse. Owners of lots to be developed in the area have submitted proposed building plans, and the builders have given you a list of building heights. It is up to you to use the heights of the proposed buildings to determine the amount of water that will accumulate between the new buildings after the first heavy rain.

The input will be a two lines, with the first line containing a number corresponding to the size of the matrix and the second line containing a space-separated list of positive integers, corresponding to the input array. The output should be a nonnegative integer corresponding to the number of units of water accumulated. Input and output processing has been handled in the sample code, if you wish to use it.

For example, given an array of proposed building heights [2, 1, 5, 2, 4], the 2D city can be drawn as such:

```
    X
    X X
    X X
X   XXX
XXXXX
```

By inspection, it is clear that water will accumulate on top of the second and fourth buildings. The expected output for this example is 3, because there is one unit of water accumulation atop the second building and 2 units of water accumulation atop the fourth building.

Sample Input	Sample Output
5 2 1 5 2 4	3

6 Balancing Act

In the near future, you have finally made it to you high school graduation ceremony, but it has been hijacked by a mad scientist who takes control of the principal's mind. Influenced by the mad scientist, the principal says he will allow the distribution of diplomas if and only if you solve a challenge: balance a scale. The mysterious object on the left-hand side of the scale has an integer mass of some number of units, and you are given a set of weights. The set of weights contains one weight for each power of 3: one 1-unit weight, one 3-unit weight, one 9-unit weight, one 27-unit weight, and so on. The mad scientist thinks that this problem is unsolvable for some values, but you realize that any integer value of mass can be balanced using this weight set. Being so convinced that the task is impossible, the mad scientist will surely expect you to balance the scale more than once, so write a program to quickly find the solution.

The input will be an integer corresponding to the mass of the mystery object on the left-hand side. The output will be a space-separated list containing only R, L, and/or -. The nth element in the list corresponds to where you should put the 3^n-unit weight. For example, if the input is 6, then the output should be this: - L R. This output corresponds to skipping the 3^0-unit (1-unit) weight, placing the 3^1-unit (3-unit) weight on the left side of the scale, and placing the 3^2-unit (9-unit) weight on the right side of the scale. The output should end with an L or an R but never a -.

Sample Input	Sample Output
6	- L R

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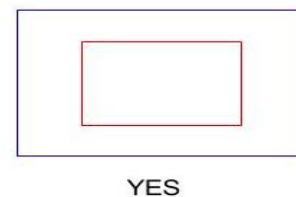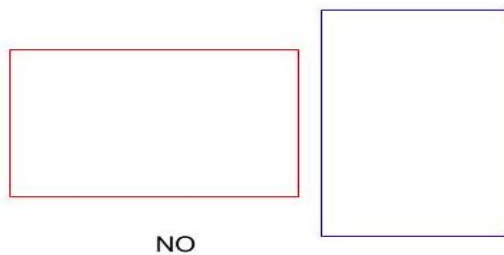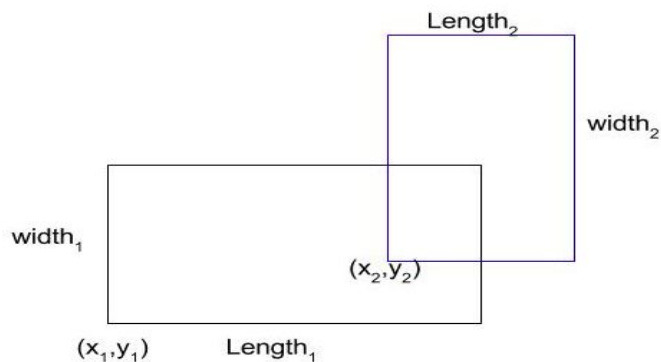
Sample Input	Sample Output
10 10 10 5 18 12 8 4	YES

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