Westford Academy Programming Club Qualifier 2013A

Each problem will have 10 test cases. The final scores will be used to choose members to attend the Fitchburg State University Programming Contest. Please submit your code to the online judge at http://wacode.github.io/. Click sign in, and make sure to enter your name exactly. Sign in prior to submission. Your score for a problem will be the most test cases ever passed on a single submission of your program. Subsequent submissions cannot lower your score, and you may resubmit as many times as you wish. Ties will be broken using the time the final score was achieved. Problems should be submitted under the origin "Qualifier" and the contest “2013A".

Be sure that your entire program fits in a single source code file. If you are using Java, make sure the file is in the default package. If you are using C++, please note that g++ will be used to compile it. If you are using Python, it will be run with the latest version. Note that there is only one correct answer for each test case. You will not be able to view the test data. Be sure your program matches the output format given in the problem exactly.

The online judge will be closed at 3:00 PM. Good luck!

Problem I. Factors (1 point per test case)

Every integer has a unique prime factoring; there are never two distinct multi-sets of prime numbers that multiply to the same integer (a multi-set is like a set but there may be identical members in a multi-set.) So the factors of twelve are the multiset {2 2 3}. This version of the definition of prime factoring is the reason that we don't count 1 as a prime, because if we did there would then be an infinite number of multisets of primes with the same product.

In this problem your program is given an integer, and it should print the smallest prime number which is a factor of the integer.

Input	Output
126	2
2431	11
262145	5

Problem II. Ordering Numbers (1 point per test case)

Given a set S_x of unique positive integers, represent that set using a range notation R_x of shortest possible size. The set notation R_x for a set S_x is either:

1. A string "N", which represents the set containing just the integer {N}
2. A string of the form "M–N" where M < N, which represents the set of consecutive integers {M, M+1, M+2, ..., N–1, N}
3. A string of the form "R_1,R_2,...,R_n", which represents the union of n individual sets S_1, S_2, ..., S_n, where the difference between the lowest integer in S_i+1 and the highest integer in S_i is at least two. Each R_i must be valid range notation. The first term for any range R_i must be less than the first term for the range R_{i+1}.

The best way to explain these rules is by example. Below are some sample sets with their corresponding range notation. Note that the only characters that appear in the range notation are: the digits "0" through "9", a "–" (dash), and "," (comma). To ensure that a range notation is the shortest possible size, you must collapse neighboring ranges. Thus, the range "1–5,6,7–10" must be represented instead as "1–10".

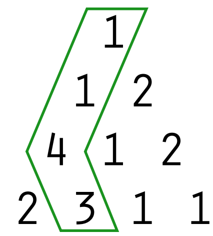
There is a special rule to which you must adhere. When a set contains just two consecutive integers, the range notation must use the dash character rather than a comma. For example, for {1, 2} the output must be "1–2" instead of "1,2"; by implication, for {1, 2, 4, 5}, the output must be "1–2,4–5".

There will be a single line of input no more than 2,000,000 characters in length. This line will contain a number of positive integers (in any order) separated by a single space between each integer.

Your output will consist of a single line representing the smallest range notation for the input set. This string will only contain the digits "0" through "9", dashes ("–") and commas (","). There must be no leading or trailing commas in the output.

Input	Output
2 6 3 5 4	2–6
2 4 3 6 7 10 9 8	2–4,6–10
99 13 15	13,15,99
1 2	1–2

Problem III: Path Counting (1 point per test case)



You are given a triangle of positive integers, such as appears on the right. The triangle is composed of N rows. The first row contains a single number; the second row contains two numbers; the third row contains three numbers, and so on. Define a path in this triangle as starting from the number on the first row and ending at a number on the Nth row. The path extends downward by either moving to the number immediately below and to the left, or below and to the right. In the example highlighted on the right, the path starts at 1 on the first row and extends to the 3 on the Nth row. The sum of the numbers in this path is 9; no other path in the above triangle has a greater sum, thus 9 is the maximal sum in this triangle. Given a triangle of numbers with N rows, write a program to compute the maximal sum of any such path. Note that each path contains N numbers.

The first line of input is a positive integer N ($1 \leq N \leq 1000$) on a line by itself representing the number of rows in the number triangle. The next N lines contain the positive integers of each successive row of the triangle, with a space between each integer. You can assume that the triangle is properly encoded in the input.

Output consists of a single positive integer on a line by itself representing the greatest sum of all such paths in the triangle.

Input	Output
3 1 2 1 1 2 3	5
4 1 1 2 4 1 2 2 3 1 1	9
5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	15