二零零七年上海交通大学计算机上机复试真题

Problem A. Old Bill
Input file: standard input
Output file: standard output

Among grandfather's papers a bill was found.
72 turkeys $_679_

The first and the last digits of the number that obviously represented the total price of those turkeys are replaced here by blanks (denoted _), for they are faded and are illegible. What are the two faded digits and what was the price of one turkey?
We want to write a program that solves a general version of the above problem.
N turkeys $_XYZ_

The total number of turkeys, N, is between 1 and 99, including both. The total price originally consisted of five digits, but we can see only the three digits in the middle. We assume that the first digit is nonzero, that the price of one turkeys is an integer number of dollars, and that all the turkeys cost the same price.
Given N, X, Y, and Z, write a program that guesses the two faded digits and the original price. In case that there is more than one candidate for the original price, the output should be the most expensive one. That is, the program is to report the two faded digits and the maximum price per turkey for the turkeys.

Input
The first line of the input file contains an integer N (0<N<100), which represents the number of turkeys. In the following line, there are the three decimal digits X, Y, and Z., separated by a space, of the original price $_XYZ_.

Output
For the input case, there may be more than one candidate for the original price or there is none. In the latter case your program is to report 0. Otherwise, if there is more than one candidate for the original price, the program is to report the two faded digits and the maximum price per turkey for the turkeys.

Sample input and output
Standard input standard output

72 6 7 9	3 2 511
5 2 3 7	9 5 18475
78 0 0 5	0

Problem B. Powerful Calculator
Input file: standard input
Output file: standard output

Today, facing the rapid development of business, SJTU recognizes that more powerful calculator should be studied, developed and appeared in future market shortly. SJTU now invites you attending such amazing research and development work.
In most business applications, the top three useful calculation operators are Addition (+), Subtraction (-) and Multiplication (×) between two given integers. Normally, you may think it is just a piece of cake. However, since some integers for calculation in business application may be very big, such as the GDP of the whole world, the calculator becomes harder to develop.
For example, if we have two integers 20 000 000 000 000 000 and 4 000 000 000 000 000, the exact results of addition, subtraction and multiplication are:
20000000000000000 + 4000000000000000 = 24 000 000 000 000 000
20000000000000000 - 4000000000000000 = 16 000 000 000 000 000
20000000000000000 × 4000000000000000 = 80 000 000 000 000 000 000 000 000 000 000 000

Note: SJTU prefers the exact format of the results rather than the float format or scientific remark format. For instance, we need "24000000000000000" rather than 2.4×10^16.
As a programmer in SJTU, your current task is to develop a program to obtain the exact results of the addition (a + b), subtraction (a - b) and multiplication (a × b) between two given integers a and b.

Input
The input file consist of two separate lines where the first line gives the integer a and the second gives b (|a| <10^200 and |b| < 10^200).

Output

For the input file, output three separate lines showing the exact results of addition (a + b), subtraction (a - b) and multiplication (a × b) of that case, one result per lines.

Sample input and output

Standard input	standard output
20000000000000000	24000000000000000
4000000000000000	16000000000000000
	80000000000000000000000000000000

Problem C. Sum of Factorials
Input file: standard input
Output file: standard output

John von Neumann, b. Dec. 28, 1903, d. Feb. 8, 1957, was a Hungarian-American mathematician who made important contributions to the foundations of mathematics, logic, quantum physics, meteorology, science, computers, and game theory. He was noted for a phenomenal memory and the speed with which he absorbed ideas and solved problems. In 1925 he received a B.S. diploma in chemical engineering from Zurich Institute and in 1926 a Ph.D. in mathematics from the University of Budapest, His Ph.D. dissertation on set theory was an important contributions to the subject.
At the age of 20, von Neumann proposed a new definition of ordinal numbers that was universally adopted. While still in his twenties, he made many contributions in both pure and applied mathematics that established him as a mathematician of unusual depth. His Mathematical Foundation of Quantum Mechanics (1932) built a solid framework for the new scientific discipline. During this time he also proved the mini-max theorem of GAME THEORY. He gradually expanded his work in game theory, and with coauthor Oskar Morgenstern he wrote Theory of Games and Economic Behavior (1944).
There are some numbers which can be expressed by the sum of factorials. For example 9, 9 = 1! + 2! + 3! . Dr. von Neumann was very interested in such numbers. So, he gives you a number n, and wants you to tell whether or not the number can be expressed by the sum of some factorials.
Well, it is just a piece of case. For a given n, you will check if there are some xi, and let n equal to

$\sum_{i=1}^{t} xi!$ ($t \geq 1$, $xi \geq 0$, $xi = xj \Longleftrightarrow i = j$)

即 $\sum_{i=1}^{t} xi!$ ($t \geq 1$, $xi \geq 0$, $xi = xj \Longleftrightarrow i = j$)

If the answer is yes, say "YES"; otherwise, print out "NO".

Input
You will get a non-negative integer n (n≤1,000,000) from input file.

Output
For the n in the input file, you should print exactly one word ("YES" or "NO") in a single line. No extra spaces are allowed.

Sample input and output

Standard input	standard output
9	YES
2	YES

Problem D. Zero-complexity Transposition
Input file: standard input
Output file: standard output

You are given a sequence of integer numbers. Zero-complexity transposition of the sequence is the reverse of this sequence. Your task is to write a program that prints zero-complexity transposition of the given sequence.

Input
The first line of the input file contains one integer n-length of the sequence ($0 < n \leqslant 10\,000$). The second line contains n integers numbers-a1, a2, ⋯, an ($-1\,000\,000\,000\,000\,000 \leqslant ai \leqslant 1\,000\,000\,000\,000\,000$).

Output
On the first line of the output file print the sequence in the reverse order.

Sample input and output

Standard input	standard output
3 1 2 3	3 2 1
5 -3 4 6 -8 9	9 -8 6 4 -3