THE UNIVERSITY OF SYDNEY

INFO5010 Programming Contest 1

Contest № 1

If not specified elsewhere the memory limit per submission is 2^{19} kB. Note that the shell which starts the compiled solution, any interpreter like the Java VM, which takes away about 300 MB, and your program have to run altogether within this limit. The time limit indicates the maximum time the execution can take before returning a time limit exceeded error.

The server test the submitted solution using:

```
exec "$@" < "$INPUT" > "$OUTPUT"
```

Languages that are allowed are:

- **Java** version 1.8.0_131 64-bit
- **Python 2** version 2.7.3
- **Python 3** version 3.2.3
- **C** gcc v4.8.1 with options -x (set compile language to C), -O2 (for level 2 optimization), -static (static link with all libraries), -lm (for link with math-library), -pipe (use pipe for communication between stages of compilation).
- **C++** g++ v4.8.1 with options -O2 (for level 2 optimization), -static (static link with all libraries), -pipe (use pipe for communication between stages of compilation).

Acknowledgements. Some of the problems have been sourced and adapted from previous contests. While we do not reveal them here, credit, authorship, intellectual property, copyright, etc, remains with the original problem authors.

Problem A: School of IT

Time limit: 2 seconds

At the School of Intellectual Testing, students are very respectful to their instructors. Students always call their instructors by their firstname but when they meet them for the first time during the day, they greet them a number of times that is proportional to their academic rank. For example, whenever a student sees Dr. Vincent, she would simply greet him "Hello Vincent". Whenever a student meets Prof. Alan, she would greet him three times:
"Hello Alan"
"Hello Alan"
"Hello Alan"

Given the rank and the firstname of the academic persons you encounter, your task it to write the right number of greetings on separate lines.

Input. An input file consists of the number N of testcases on the first line, followed by N lines. Each of these remaining N lines consists of a couple of rank r and firstname f (encoded as a single word over lower case English alphabet) separated by a blank space where $0 < r < 100$ and f contains between 1 and 10 characters. The same firstname does not appear on two different lines.

Output. The output consists of the right number of greetings on separate lines for each person in the list. Greetings for different persons should appear in the order the persons are listed in the input and each greeting for the same person should be on consecutive lines as illustrated in Table 1.

input	output
3	Hello Alan
3 Alan	Hello Alan
1 Vincent	Hello Alan
2 Bernhard	Hello Vincent
	Hello Bernhard
	Hello Bernhard

Table 1: A sample input (left) for which a correct solution prints the output (right)

Problem B: Deciding

Time limit: 3 seconds

In the olden days making decisions was easy. One would get a daisy and start to pluck its petals alternating between Do-it and Do-it-Not until the last petal was reached and a decision was made.

Gadget-man wants to use a computerised version. The idea is to work with a random string of zeros and ones. He will pick up two bits, one from each end of the string, and compare them. If they are the same (that is, both ones or both zeros), then it is Do-it. If they differ, then it is Do-it-Not. The two bits are then discarded and the process is repeated on the remaining string until all the bits are picked. The last two bits to be picked will be the decision maker. By the way, the string may be random but it always contains an even number of bits that is greater than zero.

Your task is to write a program that reads a string of zeros and ones and makes the decision for Gadget-man.

Input. The input starts with an integer N, on a line by itself, that represents the number of test cases. $1 < N < 1000$. The description for each test case consists of a string of zeros and ones. There are no blank spaces separating the bits.

Output. The output consists of a single line, for each test case, which contains a string Do-it or a string Do-it-Not.

Sample input and its output.

input	output
3	Do-it
00100010	Do-it-Not
01010101	Do-it
100001	

Problem C: Fox' Say

Time limit: 2 seconds

Determined to discover the ancient mystery—the sound that the fox makes—you went into the forest, armed with a very good digital audio recorder. The forest is, however, full of animals' voices, and on your recording, many different sounds can be heard. But you are well prepared for your task: you know exactly all the sounds which other animals make. Therefore the rest of the recording—all the unidentified noises—must have been made by the fox.

Input. The first line of input contains the number of test cases T. The descriptions of the test cases follow: The first line of each test case contains the recording—words over lower case English alphabet, separated by spaces. Each contains at most 100 letters and there are no more than 100 words. The next few lines are your pre-gathered information about other animals, in the format ⟨ animal ⟩ goes ⟨ sound ⟩. There are no more than 100 animals, their names are not longer than 100 letters each and are actual names of animals in English. There is no fox goes ... among these lines.

The last line of the test case is exactly the question you are supposed to answer: what does the fox say?

Output. For each test case, output one line containing the sounds made by the fox, in the order from the recording. You may assume that the fox was not silent (contrary to popular belief, foxes do not communicate by Morse code).

Sample input and its output.

input
1 toot woof wa ow ow ow pa blub blub pa toot pa blub pa pa ow pow toot dog goes woof fish goes blub elephant goes toot seal goes ow what does the fox say?
output
wa pa pa pa pa pa pow

Problem D: Drivers

Time limit: 5 seconds

Learner drivers need to complete 50 hours of supervised driving with at least 10 hours, or more, of night driving included in the total amount. Each learner must keep a log book of their supervised driving experience with each entry containing the starting and finishing time for each driving experience.

Poirot-the-inspector has been given the duty of checking the log books and verifying that the driving times add up to the required values and that no single driving experience exceeds 2 hours. If more than, or equal to, half the length of one driving experience occurs during the night (before sunrise or after sunset), then the whole time counts towards night driving. For example, driving from 04:50 to 06:10 on a day when sunrise is at 05:30 counts as night driving.

However, Poirot has never been good with numbers and he is requesting assistance with this duty. Your task is to write a program that reads a learner's log book and checks that the required driving times have been completed without violating the 2 hour length rule.

Input. The input consists of a number of test cases. The description for each test case starts with an integer N, on a line by itself, that represents the number of entries in one log book. 25 < N < 300. Each of the following N lines contains a record of one driving experience with four times representing sunrise, sunset, and the starting and finishing time in that order. The starting time is strictly smaller than the finishing time. A single space separates the times and each time has a value between 00:00 (midnight) and 23:59 (one minute to midnight), inclusive. A line with a zero by itself terminates the input and is not to be processed.

Output. The output for each log book consists of the string PASS if the required driving times have been completed without violating the 2 hour length rule. Otherwise, print the string NON.

Sample input and its output.

input	output
120	NON
05:34 17:41 04:01 04:18	PASS
06:49 19:02 06:27 07:29	
06:55 18:31 12:18 22:44	
...few lines deleted	
07:02 19:26 05:11 06:26	
07:01 18:16 11:30 12:51	
176	
07:36 18:33 05:00 06:24	
06:22 19:51 08:06 09:51	
...few lines deleted	
06:41 18:41 05:33 06:59	
0	