Input: Standard Input
Output: Standard Output

Problem A
An Easy Problem

Description

As we known, data stored in the computers is in binary form. The problem we discuss now is about the positive integers and its binary form. Given a positive integer I, you task is to find out an integer J, which is the minimum integer greater than I, and the number of '1's in whose binary form is the same as that in the binary form of I.

For example, if "78" is given, we can write out its binary form, "1001110". This binary form has 4 '1's. The minimum integer, which is greater than "1001110" and also contains 4 '1's, is "1010011", i.e. "83", so you should output "83".

Input

One integer per line, which is I (1 <= I <= 1000000). A line containing a number "0" terminates input, and this line need not be processed.

Output

One integer per line, which is J.

Sample Input	Sample Output
1 2 3 4 78 0	2 4 5 8 83

Input: Standard Input
Output: Standard Output

Problem B
Integer Multiplication

The problem is to multiply two integers X, Y. ($0 \quad X; Y < 10^{250}$)

Input

The input will consist of a set of pairs of lines. Each line in pair contains one multiplayer.

Output

For each input pair of lines the output line should consist one integer the product.

Sample Input	Sample Output
12 12 2 22222222222222222222222	144 444444444444444444444444

Input: Standard Input
Output: Standard Output

Problem C
Triangular wave

In this problem you are to generate a triangular wave form according to a specified pair of Amplitude and Frequency.

Input and Output

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

Each input set will contain two integers, each on a separate line. The first integer is the Amplitude; the second integer is the Frequency.

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

For the output of your program, you will be printing wave forms each separated by a blank line. The total number of wave forms equals the Frequency, and the horizontal ``height'' of each wave equals the Amplitude. The Amplitude will never be greater than nine.

The waveform itself should be filled with integers on each line which indicate the ``height'' of that line.

NOTE: There is a blank line after each separate waveform, **excluding** the last one.

Sample Input	Sample Output

1 3 2	1 22 333 22 1 1 22 333 22 1

Input: Standard Input
Output: Standard Output

Problem D
Jolly Jumper

A sequence of $n > 0$ integers is called a *jolly jumper* if the absolute values of the difference between successive elements take on all the values 1 through n 1. For instance,

1 4 2 3 is a jolly jumper, because the absolutes differences are 3, 2, and 1 respectively. The definition implies that any sequence of a single integer is a jolly jumper. You are to write a program to determine whether or not each of a number of sequences is a jolly jumper.

Input

Each line of input contains an integer n 3000 followed by n integers representing the sequence.

Output

For each line of input, generate a line of output saying 'Jolly' or 'Not jolly'.

Sample Input	Sample Output
4 1 4 2 3 5 1 4 2 -1 6	Jolly Not jolly

Input: Standard Input
Output: Standard Output

Problem E
Friends on Social Network

After IOI Ilya decided to make a business. He found a social network called "TheScorpyBook.com". It currently has **N** registered users. As in any social network two users can be friends. Ilya wants the world to be as connected as possible, so he wants to suggest friendship to some pairs of users. He will suggest user **u** to have a friendship with user **v** if they are not friends yet and there is a user **w** who is friends of both of them. Note that **u**, **v** and **w** are different users. Ilya is too busy with IPO these days, so he asks you to count how many friendship suggestions he has to send over his social network.

Input

The first line contains an integer number **N** — the number of users in the network. Next **N** lines contain **N** characters each denoting friendship relations. $\mathbf{j}^{th}$ character if the $\mathbf{i}^{th}$ lines equals one, if users **i** and **j** are friends and equals to zero otherwise. This relation is symmetric, i.e. if user **a** is friend of **b** then **b** is also a friend of **a**.

Output

Output a single integer — number of friendship suggestions Ilya has to send.

Constraints

- $1 \leq N \leq 2000$

Sample Input	Output
4 0111 1000 1000 1000	6

Input: Standard Input
Output: Standard Output

Problem F
Encryption Technique

7 PUSH 1 100 PUSH 1 200 PUSH 2 300 PUSH 2 400 POP 2 POP 1 POP 2	400 200 300

Input: Standard Input
Output: Standard Output

Problem I
Bacon's Cipher

Programmer Vasya was down on his luck. Instead of a vacation, he was sent to a scientific conference. "It is necessary to increase your competence," his boss said, "it's an important conference on cryptography, and it's held in France, where they used encryption in the days of de Richelieu and cracked codes in the days of Viete."
One of the talks at the conference was about the attempts to solve Bacon's ciphers. The speaker proposed a hypothesis that the key to Bacon's secrets could be found if all possible substrings of Bacon's works were analyzed. "But there are too many of them!" Vasya expressed his astonishment. "Not as many as you think," the speaker answered, "count them all and you'll see it yourself." That evening Vasya found on the Web the complete set of Bacon's works. He wrote a program that converted the texts into one long string by removing all linebreaks, spaces, and punctuation marks. And now Vasya is confused because he doesn't know how to calculate the number of different substrings of this string.

Input
You are given a nonempty string consisting of lowercase English letters. The string is no longer than 5000 symbols.

Output
Output the number of different substrings of this string.
Sample

input	output
aaba	8

Input: Standard Input
Output: Standard Output

Problem J
Coco-Cola

Once upon a time, there is a special coco-cola store. If you return three empty bottles to the shop, you'll get a full bottle of coco-cola to drink. If you have n empty bottles right in your hand, how many full bottles of coco-cola can you drink?
Input
There will be at most 10 test cases, each containing a single line with an integer n ($1 \le n \le 100$). The input terminates with $n = 0$, which should not be processed.
Output
For each test case, print the number of full bottles of coco-cola that you can drink.
Spoiler
Let me tell you how to drink 5 full bottles with 10 empty bottles: get 3 full bottles with 9 empty bottles, drink them to get 3 empty bottles, and again get a full bottle from them. Now you have 2 empty bottles. *Borrow another empty bottle from the shop*, then get another full bottle. Drink it, and finally return this empty bottle to the shop!
Sample Input
3
10
81
0
Sample Output
1
5
40

You have devised a new encryption technique which encodes a message by inserting between its characters randomly generated strings in a clever way. Because of pending patent issues we will not discuss in detail how the strings are generated and inserted into the original message. To validate your method, however, it is necessary to write a program that checks if the message is really encoded in the final string.

Given two strings s and t, you have to decide whether s is a subsequence of t, i.e. if you can remove characters from t such that the concatenation of the remaining characters is s.

Input

The input contains several test cases. Each is specified by two strings s, t of alphanumeric ASCII characters separated by whitespace. Input is terminated by EOF.

Output

For each test case output, if s is a subsequence of t.

Sample Input	Sample Output
sequence subsequence person compression VERDI vivaVittorioEmanueleReDiItalia caseDoesMatter CaseDoesMatter	Yes No Yes No

Input: Standard Input
Output: Standard Output

Problem G

Golden Island

Long ago there was a beautiful kingdom in the island of Sona, the golden island, deep inside Africa. The trees in Sonaisland are made of gold and farmers are the richest group of people and are also heavy tax payers.

As you know that price of gold increases every year, the minister of Sona has proposed the following tax policy.

- Pay **initTax** units of gold in the first year.
- In each of the next **slot1** years, pay one unit of gold more than the previous year.
- In each of the next **slot2** years, pay double the units of gold of the previous year.
- In each of the following years, pay number of gold units equal to the product of the number of units paid in **K** recent years.

Given an integer N, find the number of units of gold to be paid in the N^{th} year. This result can be huge, so output the result modulo 100000007 (10^8+7).

Input

First line has an integer **T** (number of test cases, $1 \le T \le 3$). Each of the next **T** lines has 5 integers, **initTax slot1 slot2 K N**.

$1 \le$ **initTax, slot1, slot2** ≤ 50

$1 \le K \le$ **slot1** + **slot2** + 1

$1 \le N \le 1000000000$ (10^9)

Output

For each test case, output the tax in units of gold to be paid in the N^{th} year modulo 100000007 (10^8+7).

Sample Input	Sample Output
3 1 3 2 4 4 1 3 2 4 7 1 3 2 4 9	4 1536 18811834

Input: Standard Input
Output: Standard Output

Problem H

Single Dimensional Array Structure

Imagine, that you are employed by a software development company. You work now on the famous "D++ project", which is devoted to the creation of a new generation programming language. Your particular task is quite prosaic, though. You are to develop the memory manager being able to work with a large number of stacks.

Input

The first line of the input contains the total number of stack operations N, $0 < N \le 100000$. Each of the next N lines contains a description of a stack operation, either in the form PUSH A B (meaning to push B into stack A), or in the form POP A(meaning to pop an element from stack A), where A is the number of stack ($1 \le A \le 1000$), and B is an integer ($0 \le B \le 10^9$). You may assume that every operation is correct (i.e., before each POP operation, the respective stack is not empty).

Output

For each POP operation, described in the input, output the value, which this POP operation gets from the top of that stack, to which it is applied. Numbers should appear according to the order of the POP operations in the input. Each number should be output in a separate line.

Sample

Sample Input	Sample Output