TASK	AVION	PARKET	ASTRO	PROSJEK	DUGOVI	HRPA

source code	`avion.pas`
	`avion.c`
	`avion.cpp`
		`parket.pas`
		`parket.c`
		`parket.cpp`
			`astro.pas`
			`astro.c`
			`astro.cpp`
				`prosjek.pas`
				`prosjek.c`
				`prosjek.cpp`
					`dugovi.pas`
					`dugovi.c`
					`dugovi.cpp`
						`hrpa.pas`
						`hrpa.c`
						`hrpa.cpp`

input	standard input (*stdin*)					
output	standard input (*stdin*)					
time limit	1 second	1 second	1 second	1 second	1 second	1 second
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	128 MB
point value	**30**	**50**	**70**	**100**	**120**	**130**
	500					

Mirko and Slavko are USKOK agents tracking the movements of an unnamed corrupt government official. Anonymous sources have tipped them about his upcoming escape attempt. They now know he plans to use his diplomatic liaisons to try and hitch a ride on a CIA blimp leaving from Severin na Kupi blimp port.

It's common knowledge that all CIA blimps have the string "**FBI**" somewhere in their registration codes. They obtained a list of all blimps scheduled for the designated day. There are **exactly** 5 blimps on the list. Write a program that will point out all CIA blimps.

INPUT

There are exactly 5 rows of input, each row representing one blimp registration code from the list. A registration code is a sequence of **at most 10** uppercase letters of the English alphabet, digits '0' to '9', or dashes '-'.

OUTPUT

The first and only line of output must contain a space separated list of integers, sorted in increasing order, indicating the corresponding input rows containing registrations of CIA blimps.

If there are no CIA blimps, output the string "HE GOT AWAY!".

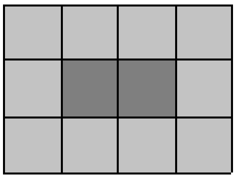
SAMPLE TESTS

input N-FBI1 9A-USKOK I-NTERPOL G-MI6 RF-KGB1 **output** 1	**input** N321-CIA F3-B12I F-BI-12 OVO-JE-CIA KRIJUMCAR1 **output** HE GOT AWAY!	**input** 47-FBI BOND-007 RF-FBI18 MARICA-13 13A-FBILL **output** 1 3 5

Ivica has set up a new parquet flooring in his room. The room is **L** decimeters long and **W** decimeters wide.

The blocks are of quadratic shape and each has an area of one quadratic decimeter. Once Ivica had set up the flooring, which consists of **brown**-colored blocks, he decided to paint the blocks **on the edge** of the room **red**.

The picture below illustrates the scenario from the test case #2 – outer blocks are red, while the remaining two inner blocks are brown:



Marica has come to visit Ivica. While Ivica was serving her cookies, she counted the number of blocks of each color. When she returned home, she recalled of the two numbers and wished to calculate the dimensions of Ivica's room. Help her!

INPUT

The first and only line of input contains two integers separated by a space, **R** (the number of red blocks) and **B** (the number of brown blocks). The following constraints will apply: $8 \le \mathbf{R} \le 5000$, $1 \le \mathbf{B} \le 2\,000\,000$.

OUTPUT

The first and only line of output must contain the dimensions of the room, **L** and **W**, respectively. If the numbers differ, output the greater one first. The test data will ensure that a unique solution always exists.

SCORING

In test cases worth 30% of total points, the dimensions of Ivica's room, **L** and **W**, will be equal.

SAMPLE TESTS

input 8 1 **output** 3 3	**input** 10 2 **output** 4 3	**input** 24 24 **output** 8 6

Ivica and Marica are attending astronomy classes and are observing two unusual stars. They noted the time when each of them flashed. They further noticed that each of the stars flashes periodically, at regular intervals, and now they are wondering: on what day, at what hour will the stars flash at the **same** minute for the **first time**?

For example, if the first star flashed today (Saturday) at 02:20 and the second one at 13:00, with the first star flashing every 05:50 (every 5 hours and 50 minutes) and the second every 01:00 (each hour), then the first one will flash again at 08:10 and 14:00, and the second one at 14:00. Therefore, both stars will have flashed at the same minute today at 14:00.

Note that the time 00:00 (midnight) is the first minute of a day.

INPUT

Four lines containing four timestamps in HH:MM format (hours:minutes), with 00 ≤ HH ≤ 23, 00 ≤ MM ≤ 59.

The timestamps are, respectively: the time of the first star's flash, the time of the second star's flash, the time between consecutive flashes of the first star, the time between consecutive flashes of the second star.

The first two timestamps will differ, and both fall on the same day - **Saturday**. Flash intervals will not be 00:00.

OUTPUT

If the stars will never flash at the same minute, output "Never" in a single line.

Otherwise, output in the first line (all lowercase) the name of the weekday of the first same-minute flash. A reminder on the correct spelling of weekdays: "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday" and "Saturday" (note the capital letter).

In the second line, output the timestamp of the first same-minute flash, in HH:MM format (with a leading zero if HH < 10).

SCORING

In test cases worth 40% of total points, both flash intervals will be hours-only (MM will equal 00), for example 13:00.

SAMPLE TESTS

input	input	input
02:20 13:00 05:50 01:00	02:20 23:28 00:40 23:50	23:19 10:19 02:42 09:11
output	**output**	**output**
Saturday 14:00	Never	Thursday 00:31

Slavko decided to challenge Mirko! He gave him a real number **P** and a bag full of pieces of paper with exactly one number 1-5 written on each paper. There is an unlimited quantity of each type of paper.

Mirko's task is to pick **the minimum number of papers** in a way that the average of the numbers written on them equals exactly **P**.

INPUT

First and only line of input contains real number **P**.

P will have between 1 and 9 decimal places, inclusive (1 ≤ **P** ≤ 5).

OUTPUT

First and only line of output should contain five nonnegative integers - numbers of ones, twos, threes, fours and fives used, respectively. **If there are multiple solutions, output any one of them.**

SAMPLE TESTS

input 5.0 **output** 0 0 0 0 1	**input** 4.5 **output** 0 0 0 1 1	**input** 3.20 **output** 0 0 4 1 0

In a little town called Križ live **N** people. Each of them has borrowed some money from **exactly one** other inhabitant. Now the time has come to pay back all the debts, but the problem is that everybody has spent all of their money!

The major of Križ has decided to solve this problem. The town will give money to a few people so that they can pay back their debts. When some people get their money back, a chain reaction is started - for example: person A gets money from the city. Person A uses that money to pay the debt toward person B. Person B then uses that money to pay the debt towards person C etc. If person B didn't have enough money to pay back the debt, they wait until they get enough. If they have more than enough money, person B will keep what is left after payback.

Another example: if two people live in Križ, and they owe $100 to each other, the town will give $100 to one of them so they can pay back the debt to the other one.

Your task is to calculate the **minimum total amount of money** the town has to give to some subset of the inhabitants so that after the payback protocol described above all debts are payed.

INPUT

First line of input contains one integer **N** ($2 \le N \le 200\ 000$), number of inhabitants of Križ. They are numbered from **1** to **N**.

The following **N** lines contain two integers, separated by space. In **i**-th of those lines, first number - **Ai** represents the id of the person **i**-th person owes money to ($1 \le A_i \le N$, $A_i \ne i$), and second **Bi** represents the ammount of the debt in \$ ($1 \le B_i \le 10\ 000$).

OUTPUT

First and only line of output should contain one integer - the minimum total ammount of money town has to give to its inhabitants so all debts are returned.

SAMPLE TESTS

input	input	input
4 2 100 1 100 4 70 3 70	3 2 120 3 50 2 80	5 3 30 3 20 4 100 5 40 3 60
output	**output**	**output**
170	150	110

Mirko and Slavko's favourite pastime is competing against each other in mathematical games. This time they took a heap of **N** pebbles and settled on the following rules:

1. Mirko is the first to play, then Slavko, then Mirko again, then Slavko and so on;
2. Mirko can take any number of pebbles (between 1 and **N**, inclusive) from the heap during his first move;
3. In each of the following turns the current player must take at least 1 pebble and is allowed to take **at most double the amount of pebbles taken during the previous turn** by the other player; naturally, one cannot take more pebbles than the remaining amount in the heap;
4. The player who takes the last pebble is the winner.

Both Mirko and Slavko play optimally (if it is possible for one player to beat the other, that player will always win). We need to find the **minimum** number of pebbles that Mirko must take during his first turn such that he is guaranteed to win the game.

INPUT

The first and only line of input contains the positive integer **N** ($2 \le \mathbf{N} \le 10^{15}$), the number of pebbles in the starting heap.

OUTPUT

The first and only line of output must contain the required minimum number of pebbles that Mirko needs to remove during his first turn.

SAMPLE TESTS

input 4 **output** 1	**input** 7 **output** 2	**input** 8 **output** 8

First sample description:

Mirko has 4 possibilities to choose from: he can take 1, 2, 3, or 4 pebbles from the heap. If he takes all 4 pebbles he will naturally win, but that is not the minimum solution. We need to check the remaining alternatives. If Mirko takes only one pebble, Slavko is left with a heap of 3, but he can take at most 2. Slavko cannot take all pebbles, but Mirko will be able to take all remaining pebbles during his next turn, winning the game. We conclude that 1 is the minimum solution for this test case.