Telerik Software Academy 2011 / 2012 – C# Fundamentals Part 1 – Test Exam

Problem 1 – Math Expression

You are given the following mathematical expression:

$$\frac{N^2 + \frac{1}{M * P} + 1337}{N - 128.523123123 * P} + \sin(\text{M mod } 180)$$

The **sin(x)** is a trigonometric function that returns the sine from the angle **x** (measured in radians)**.**

The **mod** operator finds the remainder of division of one number by another.

Here are some examples for how the **mod** operator should work:

- 5 mod 2 = 1
- 5.99 mod 3 = 2
- 6 mod 3 = 0

Your task is to write a computer program that calculates the result from the shown mathematical expression, depending on the values of the variables **N**, **M** and **P**.

Input

The input data is being read from the console.

The input consists of exactly 3 lines. In each line you consequently enter the variables **N**, **M** and **P**.

The separator between the integer and the fractional part of the number is “.” (dot).

The number of digits that follow the decimal point will not be more than 6.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data must be printed on the console.

There must be only one line, showing the result from the mathematical expression.

The result must show exactly 6 digits after the “.” (decimal point).

Constraints

- The numbers **N, M** and **P** are fractional numbers.
- **N, M** and **P** will be between -10 000 000 and 10 000 000, inclusive.
- The numbers **M** and **P** will always have values other than 0
- It is guaranteed that none of the combinations of the numbers **N, M** and **P** will lead to dividing by zero.
- Allowed working time for your program: 0.10 seconds.
- Allowed memory: 16 MB.

Examples

Input Examples	Output Examples
1 2 3	-2.570352
0.1234 1.2345 2.3456	-3.596568
0.123456 1.234567 2.345678	-3.596421

Problem 2 – Least Majority Multiple

Given five positive integers, their least majority multiple is the smallest positive integer that is divisible by **at least three** of them.

Your task is to write a program that for given distinct integers **a**, **b**, **c**, **d** and **e**, returns their least majority multiple.

For example if we have 1, 2, 3, 4 and 5 the majority multiple of the given five numbers is 4 because it is divisible by 1, 2, and 4.

Another example: if we have 30, 42, 70, 35 and 90 the answer will be 210, because it is divisible by 30, 42, 70, and 35 - four out of five numbers, which is a majority.

Input

The input data is being read from the console.

The input data will consist of 5 lines.

The numbers **a**, **b**, **c**, **d** and **e** will each be on a single line.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data must be printed on the console.

On the only output line you must print the least majority multiple of the given numbers.

Constraints

- **a**, **b**, **c**, **d** and **e** will each be integer numbers between 1 and 100, inclusive.

- **a**, **b**, **c**, **d** and **e** will be distinct.
- Allowed working time for your program: 0.25 seconds.
- Allowed memory: 16 MB.

Examples

Input Examples	Output Examples
1 2 3 4 5	4
30 42 70 35 90	210

Problem 3 – Trapezoid

Write a program that prints on the console the border of a trapezoid by given number **N**.

The width of the top side of the trapezoid must be exactly **N**.

The width of the bottom side of the trapezoid must be exactly **2 * N**.

The height of the trapezoid must be exactly **N + 1**.

Also the top right and the bottom right angle of the trapezoid must be equal to 90 degrees.

See the examples bellow.

Input

The input data is being read from the console.

On the only line in the console you are given an integer number **N**, showing the width of the smallest trapezoid side.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data must be printed on the console.

You must write the border of the described trapezoid on the console.

Use the symbol “*****” (asterisk) to mark the border of the trapezoid.

Use the symbol “.” (dot) to illustrate the empty spaces outside and inside the trapezoid.

Constraints

- The number **N** is a positive integer between 3 and 39, inclusive.
- Allowed working time for your program: 0.25 seconds.
- Allowed memory: 16 MB.

Examples

Input Examples	**Output Examples**
5	.....***** ....*....* ...*.....* ..*......* .*.......* **********
10	..........********** ....*.........* ....*..........* ....*...........* ......*............* .....*.............* ....*..............* ...*...............* ..*................* .*.................* ********************

Problem 4 – Odd Number

You are given a list of **N** integer numbers all but one of which appears an even number of times.

Write a program to find the one integer which appears an **odd number of times**.

Input

The input data is being read from the console.

The number **N** is written on the first input line.

On each of the following **N** lines there is one integer number written – the consequent number from the given list of numbers.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data must be printed on the console.

On the only output line you must print the integer from the list which appears an odd number of times.

Constraints

- **N** will be positive odd integer number between 1 and 99 999, inclusive.
- All of the numbers in the list will be integer numbers between -9 223 372 036 854 775 808 and 9 223 372 036 854 775 807, inclusive.
- Always only one answer will exists and will be unambiguous.
- Allowed working time for your program: 0.25 seconds.
- Allowed memory: 16 MB.

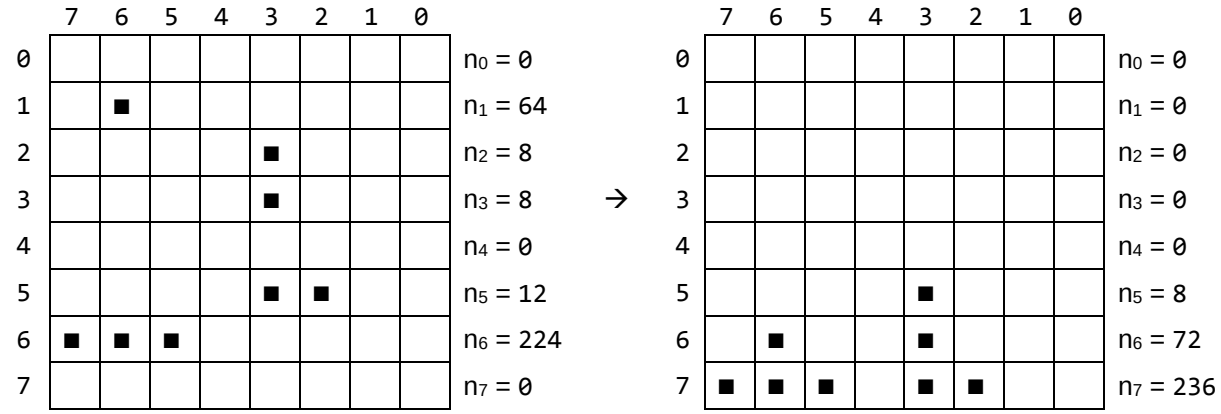
Examples

Input Examples	**Output Examples**
1 2	2
3 2 -1 2	-1
9 -1 0 1 2 3 2 1 0 -1	3

13 -1 7 7 -9223372036854775808 7 -9223372036854775808 -3 7 0 -1 7 0 -3	7

Problem 5 – Fall Down

You are given a list of **8** bytes (positive integers in the range [0…255]) $\mathbf{n_0}$, $\mathbf{n_1}$, …, $\mathbf{n_7}$. These numbers represent a square grid consisting of **8** lines and **8** columns. Each cell of the grid could either be empty or full. The first line is represented by the bits of $\mathbf{n_0}$, the second – by the bits of $\mathbf{n_1}$ and so on, and the last line is represented by the bits of $\mathbf{n_7}$. Each bit with value 1 denotes a full cell and each bit with value 0 denotes an empty cell. The lines are numbered from the first (top) to the last (bottom) with the numbers 0, 1, …, 7. The columns are numbered from right to left with the indices 0, 1, …, 7. The figure shows a sample square grid and its representation by a sequence of 8 numbers $\mathbf{n_0}$, $\mathbf{n_1}$, …, $\mathbf{n_7}$:



Suppose the full cells hold squares which can "fall down" by the influence of the gravity. Each full cell in certain row and column falls down to the lowest row possible but stays in the same column and up from any other full cells on the same column that ware initially down from it. At the figure the "fall down" process is illustrated.

Write a program to calculate how the grid will look like after the "fall down" process is applied.

Input

The input data is being read from the console.

There will be exactly 8 lines each holding the integer numbers n_0, n_1, …, n_7.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output consists of the numbers n_0, n_1, …, n_7 after the "fall down process".

Ouput should be printed on the console, in exactly 8 lines, each holding a single integer.

Constraints

- The numbers n_0, n_1, …, n_7 are positive integers between 0 and 255, inclusive.
- Allowed work time for your program: 0.25 seconds.
- Allowed memory: 16 MB.

Examples

Input Example	Output Example
0 64 8 8 0 12 224 0	0 0 0 0 0 8 72 236
255 255 255 255 255 255 255 254	254 255 255 255 255 255 255 255