Tower of Hanoi

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

The **Tower of Hanoi** is a mathematical game or puzzle consisting of three rods and a number of disks of various diameters, which can slide onto any rod. The puzzle begins with the disks stacked on one rod in order of decreasing size, the smallest at the top, thus approximating a conical shape. The objective of the puzzle is to move the entire stack to the last rod, obeying the following rules:

1. Only one disk may be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
3. No disk may be placed on top of a disk that is smaller than it.

With 3 disks, the puzzle can be solved in 7 moves. The minimal number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks, and it has been proved that there exists **only one** solution with this minimal number of moves.

Your task is to output all moves of this unique solution with minimal number of moves.

You can read Wikipedia for help: http://en.volupedia.org/wiki/Tower_of_Hanoi

Input

One integer: the number of disks n.

Output

You should print all moves by the following format:

Rods are numbered from 1 to 3. At the beginning all disks are on rod 1. finally all disks should be on rod 3.

Each step prints like this: "1->2"(without quotes)

Sample Test Data

```
1
<|==|>
1->3
```

```
3
<|==|>
1->3
1->2
3->2
1->3
2->1
2->3
1->3
```

Tips

The format of your output should be the same as the sample outputs.

This problem is a famous recursive problem, so please try to use recursive algorithms to practise.

Data Limit

For 50% cases: $1 <= n <= 8$

For 100% cases: $1 <= n <= 18$

Attention

The output can be huge, and you can reduce input time by using fast output.

C++: using **putchar()** instead of **printf()**

Java fast I/O: https://paste.ubuntu.com/p/6ybMcVXvz5/

Zero Point

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	5	Tradition

Description

There is a function:

$$f(x) = (Ax^2 + Bx + C)e^x - \pi^{1.14514}$$

It is guaranteed that this function has one and only one zero point within $(-10, 10)$.

Please find out this zero point.

Input

Three decimals: A, B, C.

Output

One decimal, representing the zero point.

answer should keep four decimal places.

Sample Test Data

```
10 3 7
<|==|>
-3.4264
```

```
12.34 56.78 -9.99
<|==|>
0.2181
```

Tips

Data Limit

For 100% cases: $-100 <= A, B, C <= 100$

Attention

Crazy Function

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

There is a discrete function $W(A, B, C, D)$. The recursive definition is as follow:

1.$W(1, 1, 1, 1) = 1$;

2.if $A, B, C, D > 1$, then

$W(A, B, C, D) = W(A - 1, B, C, D) + W(A, B - 1, C, D) + W(A, B, C - 1, D) + W(A, B, C, D - 1)$
;

3.if some variable equals to 1, for example:

if $A = 1$, then
$W(A, B, C, D) = W(A, B - 1, C, D) + W(A, B, C - 1, D) + W(A, B, C, D - 1)$;

if $A = B = 1$,then $W(A, B, C, D) = W(A, B, C - 1, D) + W(A, B, C, D - 1)$;

and so on.

We can easily calculate this function by recursive way. But the problem is, the recursive calculation will be very slow if A, B, C, D are all exceed 5. Now please solve it by some faster way.

Input

Four integers: A, B, C, D.

Output

One integer: $W(A, B, C, D)$.

Since the answer may be larger than $2^{63} - 1$, we can only print the answer MOD 98244353.

Sample Test Data

```
1 1 2 3
<|==|>
3
```

```
5 6 6 6
<|==|>
936697550
```

Tips

For the first case, $W(1, 1, 2, 3) = W(1, 1, 2, 2) + W(1, 1, 1, 3)$

$= W(1, 1, 1, 2) + W(1, 1, 2, 1) + W(1, 1, 1, 2)$

$= W(1, 1, 1, 1) + W(1, 1, 1, 1) + W(1, 1, 1, 1) = 3$.

For the second case, the answer is 2933186256 MOD 998244353 = 936697550.

Data Limit

For 50% cases: $A, B, C, D <= 5$

For 100% cases: $1 <= A, B, C, D <= 30$

Attention

Cutting Wood

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

The lumber factory has n logs and now wants to cut these logs into k pieces with **equal** length l.

We want to get as long pieces of wood as possible, so we're asking for the maximum of l.

The unit of wood length is cm. The length of logs is a positive integer, and the length of small pieces is also a positive integer.

For example, there are two logs with lengths of 11 and 21 respectively, which are required to be cut into 6 pieces of equal length. Obviously, the longest length of small pieces of wood that can be cut out is 5.

Input

The first line contains two positive integers n, k, representing the number of logs and the number of pieces needed.

The next line, contains n positive integer L_i, representing the length of a log.

Output

One line, containing the maximum of l.

If it can't be cut to pieces of even $1cm$, print 0.

Sample Test Data

```
2 6
11 21
<|==|>
5
```

```
3 7
232
124
456
<|==|>
114
```

Tips

Data Limit

For 50% cases: $n, L_i <= 1000$

For 100% cases: $1 <= n <= 100000, 1 <= k, Li <= 10^9$

Attention

Remember: Use **long** instead **int**

Product

Information

Time Limit	Memory Limit	Data Amount	Problem Type
2000ms	128MiB	10	Tradition

Description

There are two arrays: $a_1, ..., a_n$ and $b_1, ..., b_m$. If we multiply the elements of a times the elements of b, there will be $n * m$ numbers: $a_i * b_j (1 <= i <= n, 1 <= j <= m)$.

Please find out the k_{th} smallest number of $a_i * b_j$.

Input

The first line contains three integers n, m, k.

The next line contains n integers: $a_1, ..., a_n$.

The next line contains m integers: $b_1, ..., b_m$.

It is guaranteed that $a_1, ..., a_n$ and $b_1, ..., b_m$ has been sorted in non-decreasing order.

Output

One line, containing the k_{th} smallest number.

Sample Test Data

```
4 4 6
1 2 2 5
2 3 4 6
<|==|>
6
```

```
4 4 11
1 2 2 5
2 3 4 6
<|==|>
10
```

Tips

All products:

a_i/b_j	**2**	**3**	**4**	**6**
1	2	3	4	6
2	4	6	8	12
2	4	6	8	12
5	10	15	20	30

after sorting: 2 3 4 4 4 6 6 6 8 8 10 12 12 15 20 30

It can be seen that the 6_{th} smallest is 6, and the 11_{th} smallest is 10.

Data Limit

For 50% cases: $n, m <= 1000$

For 100% cases: $1 <= n, m <= 100000, 1 <= a_i, b_j <= 10^9, 1 <= k <= n * m$

Attention