A. A + B

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two integers A and B. Calculate their sum and output it without leading zeros.

Input

Two lines of input data contain integers A and B $(1 \le A, B \le 10^5)$.

Output

Output $A + B$ without leading zeros.

Sample test(s)

input

```
12
3
```

output

```
15
```

input

```
100
5
```

output

```
105
```

Note

The code provided in the post about the round doesn't solve the task.

B. Binary notation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a positive integer n. Output its binary notation.

Input

The only line of input data contains an integer n ($1 \le n \le 10^6$).

Output

Output the binary notation of n (without any leading zeros).

Sample test(s)

input

```
5
```

output

```
101
```

input

```
13
```

output

```
1101
```

Note

In the first example $5 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0$.

C. Caesar Cipher

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Caesar cipher is one of the simplest encryption techniques. To transform the original message into encrypted one using key k, one has to replace each letter with a letter which is k positions later in the alphabet (if this takes the position beyond Z, the rest of it is counted from the start of the alphabet). In a more formal way, if letters of the alphabet are enumerated starting with 0, the result of encryption for character x will be $(x+k) \mod 26$ (26 is the number of letters in the Latin alphabet).

You are given the original message and the encryption key k. Output the resulting cipher.

Input

The first line of input contains the original message — a sequence uppercase Latin letters («`A`»-«`Z`»). The length of the message is from 1 to 10, inclusive.

The second line contains an integer k ($0 \leq k \leq 25$).

Output

Output the result of encryption.

Sample test(s)

input

```
CODEFORCES
5
```

output

```
HTIJKTWHJX
```

input

```
WIXYZILWYM
6
```

output

```
CODEFORCES
```

D. Date Change

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a date in "`DD.MM.YYYY`" ("day.month.year") format and a number of days $shift$ you have to add to this date. Output the resulting date.

Input

The first line of input contains the date in "`DD.MM.YYYY`" format: two digits for day (with leading zero if needed), dot, two digits for month (with leading zero if needed), dot, four digits for year. The notation is guaranteed to give a valid date between 1980 and 2020, inclusive.

The second line contains an integer $shift$ ($-1000 \le shift \le 1000$).

Output

Output a date equal to the given one + $shift$ days, in the same format "`DD.MM.YYYY`".

Sample test(s)

input

```
10.02.2012
12
```

output

```
22.02.2012
```

input

```
01.02.2010
-40
```

output

```
23.12.2009
```

input

```
01.01.2000
365
```

output

```
31.12.2000
```

input

```
13.08.1990
-609
```

output

```
12.12.1988
```

Note

When manipulating the dates, take into account leap years; don't care about time zones/daylight saving time.

E. Euclidean Distance

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a multiset of points on the plane with integer coordinates. Find the maximum distance between two points from this multiset.

Input

The first line of input contains the number of points n ($2 \le n \le 50$). Next, n pairs of lines follow, each describing a single point: the first line contains x-coordinate, the second one — the y-coordinate ($-50 \le x, y \le 50$). Some of the points can have identical coordinates.

Output

Output the maximum distance between two points from this multiset. The answer is considered to be correct if its absolute or relative error does not exceed 10^{-4}.

Sample test(s)

input

```
3
0
1
2
3
4
5
```

output

```
5.656854249
```

input

```
3
10
12
-5
8
10
12
```

output

```
15.5241747
```

Note

In the first case the maximum distance is between points $(0, 1)$ and $(4, 5)$. In the second case two of the points are the same, so the maximum distance is between one of them and the third point.