A. Eevee

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are solving the crossword problem K from IPSC 2014. You solved all the clues except for one: who does Eevee evolve into? You are not very into pokemons, but quick googling helped you find out, that Eevee can evolve into eight different pokemons: Vaporeon, Jolteon, Flareon, Espeon, Umbreon, Leafeon, Glaceon, and Sylveon.

You know the length of the word in the crossword, and you already know some letters. Designers of the crossword made sure that the answer is unambiguous, so you can assume that exactly one pokemon out of the 8 that Eevee evolves into fits the length and the letters given. Your task is to find it.

Input

First line contains an integer n ($6 \le n \le 8$) – the length of the string.

Next line contains a string consisting of n characters, each of which is either a lower case english letter (indicating a known letter) or a dot character (indicating an empty cell in the crossword).

Output

Print a name of the pokemon that Eevee can evolve into that matches the pattern in the input. Use lower case letters only to print the name (in particular, do not capitalize the first letter).

Examples

input

```
7
j......
```

output

```
jolteon
```

input

```
7
...feon
```

output

```
leafeon
```

input

```
7
.l.r.o.
```

output

```
flareon
```

Note

Here's a set of names in a form you can paste into your solution:

["vaporeon", "jolteon", "flareon", "espeon", "umbreon", "leafeon", "glaceon", "sylveon"]

{"vaporeon", "jolteon", "flareon", "espeon", "umbreon", "leafeon", "glaceon", "sylveon"}

B. 4-point polyline

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

You are given a rectangular grid of lattice points from $(0, 0)$ to (n, m) inclusive. You have to choose exactly 4 different points to build a polyline possibly with self-intersections and self-touching. This polyline should be as long as possible.

A polyline defined by points p_1, p_2, p_3, p_4 consists of the line segments $p_1\,p_2, p_2\,p_3, p_3\,p_4$, and its length is the sum of the lengths of the individual line segments.

Input

The only line of the input contains two integers n and m $(0 \le n, m \le 1000)$. It is guaranteed that grid contains at least 4 different points.

Output

Print 4 lines with two integers per line separated by space — coordinates of points p_1, p_2, p_3, p_4 in order which represent the longest possible polyline.

Judge program compares your answer and jury's answer with 10^{-6} precision.

Examples

input

```
1 1
```

output

```
1 1
0 0
1 0
0 1
```

input

```
0 10
```

output

```
0 1
0 10
0 0
0 9
```

C. Magic Trick

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alex enjoys performing magic tricks. He has a trick that requires a deck of n cards. He has m identical decks of n different cards each, which have been mixed together. When Alex wishes to perform the trick, he grabs n cards at random and performs the trick with those. The resulting deck looks like a normal deck, but may have duplicates of some cards.

The trick itself is performed as follows: first Alex allows you to choose a random card from the deck. You memorize the card and put it back in the deck. Then Alex shuffles the deck, and pulls out a card. If the card matches the one you memorized, the trick is successful.

You don't think Alex is a very good magician, and that he just pulls a card randomly from the deck. Determine the probability of the trick being successful if this is the case.

Input

First line of the input consists of two integers n and m ($1 \le n, m \le 1000$), separated by space — number of cards in each deck, and number of decks.

Output

On the only line of the output print one floating point number – probability of Alex successfully performing the trick. Relative or absolute error of your answer should not be higher than 10^{-6}.

Examples

input

```
2 2
```

output

```
0.6666666666666666
```

input

```
4 4
```

output

```
0.4000000000000000
```

input

```
1 2
```

output

```
1.0000000000000000
```

Note

In the first sample, with probability $\frac{1}{3}$ Alex will perform the trick with two cards with the same value from two different decks. In this case the trick is guaranteed to succeed.

With the remaining $\frac{2}{3}$ probability he took two different cards, and the probability of pulling off the trick is $\frac{1}{2}$.

The resulting probability is $\frac{1}{3} \times 1 + \frac{2}{3} \times \frac{1}{2} = \frac{2}{3}$

D. Washer, Dryer, Folder

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have k pieces of laundry, each of which you want to wash, dry and fold. You are at a laundromat that has n_1 washing machines, n_2 drying machines and n_3 folding machines. Each machine can process only one piece of laundry at a time. You can't dry a piece of laundry before it is washed, and you can't fold it before it is dried. Moreover, after a piece of laundry is washed, it needs to be immediately moved into a drying machine, and after it is dried, it needs to be immediately moved into a folding machine.

It takes t_1 minutes to wash one piece of laundry in a washing machine, t_2 minutes to dry it in a drying machine, and t_3 minutes to fold it in a folding machine. Find the smallest number of minutes that is enough to wash, dry and fold all the laundry you have.

Input

The only line of the input contains seven integers: $k, n_1, n_2, n_3, t_1, t_2, t_3$ ($1 \le k \le 10^4$; $1 \le n_1, n_2, n_3, t_1, t_2, t_3 \le 1000$).

Output

Print one integer — smallest number of minutes to do all your laundry.

Examples

input

```
1 1 1 1 5 5 5
```

output

```
15
```

input

```
8 4 3 2 10 5 2
```

output

```
32
```

Note

In the first example there's one instance of each machine, each taking 5 minutes to complete. You have only one piece of laundry, so it takes 15 minutes to process it.

In the second example you start washing first two pieces at moment 0. If you start the third piece of laundry immediately, then by the time it is dried, there will be no folding machine available, so you have to wait, and start washing third piece at moment 2. Similarly, you can't start washing next piece until moment 5, since otherwise there will be no dryer available, when it is washed. Start time for each of the eight pieces of laundry is $0, 0, 2, 5, 10, 10, 12$ and 15 minutes respectively. The last piece of laundry will be ready after $15 + 10 + 5 + 2 = 32$ minutes.

E. Three strings

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

You are given three strings (s_1, s_2, s_3). For each integer l $(1 \le l \le min(|s_1|, |s_2|, |s_3|)$ you need to find how many triples (i_1, i_2, i_3) exist such that three strings $s_k[i_k \dots i_k + l - 1]$ $(k = 1, 2, 3)$ are pairwise equal. Print all found numbers modulo 1000000007 $(10^9 + 7)$.

See notes if you are not sure about some of the denotions used in the statement.

Input

First three lines contain three non-empty input strings. The sum of lengths of all strings is no more than $3 \cdot 10^5$. All strings consist only of lowercase English letters.

Output

You need to output $min(|s_1|, |s_2|, |s_3|)$ numbers separated by spaces — answers for the problem modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
abc
bc
cbc
```

output

```
3 1
```

input

```
abacaba
abac
abcd
```

output

```
11 2 0 0
```

Note

Consider a string $t = t_1 t_2 \dots t_{|t|}$, where t_i denotes the i-th character of the string, and $|t|$ denotes the length of the string.

Then $t[i \dots j]$ $(1 \le i \le j \le |t|)$ represents the string $t_i t_{i+1} \dots t_j$ (substring of t from position i to position j inclusive).

F. Permutation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a permutation of numbers from 1 to n. Determine whether there's a pair of integers a, b $(1 \le a, b \le n;\ a \ne b)$ such that the element $\frac{(a+b)}{2}$ (note, that it is usual division, not integer one) is between a and b in this permutation.

Input

First line consists of a single integer n $(1 \le n \le 300000)$ — the size of permutation.

Second line contains n integers — the permutation itself.

Output

Print "YES", if such a pair exists, "NO" otherwise (in both cases without quotes, the answer is case insensitive).

Examples

input

```
4
1 3 4 2
```

output

```
NO
```

input

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

output

```
YES
```

Note

In the second example 2 is between 1 and 3. Additionally 4 is between 3 and 5.