A – A Tour of Home!

Filename: tour

You've finally arrived in Miami, at last! Since it has been a while since you've last seen Henry, he would like to show you special sites around Miami to see what's changed, and so he has come up with a list of n places he's *very* eager to show you. Unfortunately, due to the chronic construction going on in the downtown area, there will only be m one-way roads available to you guys to travel around the city. It is therefore possible that not all the special sites can be visited. Although Henry is a pro at figuring out the hot spots worth checking out, the restriction of roads has him very stirred up, so he can't think straight! However, he insists that even though the sites may be visited in any order, no site may be visited more than once (except for the unspecified start point). Since he does not remember the tour he originally came up with, any ordering you give him will be considered valid as long as it satisfies this requirement. However, since you happen to be an expert programmer, you tell him to leave it to you! Please help Henry out in these troubling times of construction chaos!

Input

The first line of input will contain two integers n and m, $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$: the number of sites, and the number of roads between sites, respectively. The next m lines of input will each contain two integers $0 \leq a \neq b < n$ denoting that a road goes from a to b. Each of these roads will only show up once in the input.

Output

The first line of output should contain a single integer k, denoting the number of places visited on the new itinerary; note that the starting site should be equal to the ending site. The second line should contain k integers, each separated by a single space, denoting the sites in the itinerary from beginning to end. If no such trip is doable, output `IMPOSSIBLE` on a single line.

Sample Input 1

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

Sample Output 1

```
4
1 3 2 1
```

Sample Input 2

```
6 6
1 2
1 3
1 4
2 5
4 5
5 6
```

Sample Output 2

```
IMPOSSIBLE
```

B – Classroom Conundrum

Filename: conundrum

The spring semester is in full swing here at FIU, and Nicholas has a particularly heavy course load this time around! This is slightly problematic since this semester, he is also interning at the Los Alamos National Laboratory on a *very* secret project (don't tell anyone he told you!) he's excited about the **H**igh-**S**peed **P**article **C**ollider! Because of his internship, he finds himself dangerously close to not being able to pass one of his classes! Luckily, his professor, Dr. Giri, is letting students work in teams for the midterm! However, the midterm has a number of restrictions, as follows:

1. All the students will be taking a midterm in a straight line. The order in which students will appear is arbitrarily decided by Dr. Giri.

2. In order for students to group up, they must be next to each other on this line: for example, if three students wanted to form a group, these three students must be next to each other on a line.

Dr. Giri has released the order in which students will sit in this line from first to last, but it is **encoded**: instead of each student's name, each student's rank in the class so far is used in place of their name, where a higher number indicates better performance. Nicholas doesn't know exactly what his rank is, and so he has no idea where in this list he actually is. In light of this alarming information, he has asked you the following: for each possible position i in this list, assuming Nicholas is at position i, how many groups could he possibly form, given that he must have the lowest of all class rankings in his own group (he wants everyone to be smarter than him so that they may carry him to success)?

Input

The first line of input will contain a single integer $n, 1 \leq n \leq 10^5$: the number of students taking the midterm (what a large class size!). The next line of input contains the sequence of n integers, each separated by a single space, denoting the rankings of the students as given by Dr. Giri's ordering. Each student's rank is **unique** and is some number between 1 and n, inclusive.

Output

There should be only one line of output containing a space-separated list of n integers, the ith of which reports the number of groups Nicholas could form assuming he is at position i. Be mindful of the magnitude of your results!

Sample Input 1

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

Sample Output 1

```
1 4 1 4 5
```

Sample Input 2

```
3
1 3 2
```

Sample Output 2

```
3 1 2
```

C – Digits, digits, digits!

Filename: digits

Monica has recently taken a keen interest in number theory, on account of her already fierce passion for digits! Because of this passion for digits, one class of problems immediately catches her eyes: digit sum problems! The digit sum of a number is simply the sum of all of the digits of this number. For example, the digit sum of 1379 is $1 + 3 + 7 + 9$, which is 20. One of the problems in her number theory textbook has her puzzled, and she's been staying up at night trying to figure out the solution! It reads as follows:

> Given an integer s, which is the digit-sum of *some* positive number k, how many possible values of k are there, assuming that k has no zeroes in its digit representation at all?

After much thought, she has decided that she is not sufficiently prepared to approach this problem, and so she would like you, an esteemed number theorist, to help her solve this treacherous problem!

Input

The first line of input will contain a singler integer s, $1 \leq s \leq 200$, as described by the textbook question.

Output

There should only be a single line of output, containing a single integer: the answer to this question; as this number can be fairly large, she has asked you to instead give the integer modulo $10^9 + 7$.

Sample Input 1

```
1
```

Sample Output 1

```
1
```

Sample Input 2

```
5
```

Sample Output 2

```
16
```

D – Geometric Gallivanting

Filename: geometric

Ricky has decided to visit Stonehenge, a famous pre-historic monument located in England! Even though there are instructions against going near it, Ricky *really* wants to stand in the center of all the stone blocks! He's not very good at eyeballing things, so he needs your help in writing a program that takes the locations of the stones (which all lie on a flat plane) and outputs the geometric center of these stones. He wants you to make two simplifying assumptions: all the stones have the same mass, and their locations can be treated as points.

Input

The first line of input will contain a single integer $n \leq 10^5$, denoting the number of stones. Following this is n lines of input, each with a pair of integers x and y, $-10^8 \leq x, y \leq 10^8$, separated by a single space; the ith of these lines denotes the point location of the ith stone.

Output

The first and only line of output should contain two real numbers separated by a single space, denoting the center of the stones. The location printed will be judged as correct if it is at a distance less than 10^{-3} units from the true center.

Sample Input 1

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

Sample Output 1

```
2.500000 2.500000
```

Sample Input 2

```
2
1 0
-2 0
```

Sample Output 2

```
-0.500000 0.000000
```

E – Something Special

Filename: special

It's another beautiful day of the Spring semester here at FIU, for which Ousman is very thankful! He is so thankful, in fact, that he's going to take this opportunity to appreciate one of the campus's hottest spots: the lake in front of the Green Library! When he got to the lake, he noticed a not-uncommon sight: turtles! But there was *something* special about the turtles he saw: they happened to be in a line! Ousman decided he was going to keep one of these turtles as a pet, but he must first consider three things about each turtle: how hungry it is, how obedient it is, and how large it is. He wants the turtle with maximum *docility* – a property Ousman defines as being obedience times hunger minus size. This means that if the ith turtle has obedience o_i, hunger h_i and size s_i, its docility is equal to $o_i \cdot h_i - s_i$. If there are several such turtles, he will take the one closest to the beginning of the line. Given the hunger, obedience, and size of each of the turtles, can you tell Ousman which turtle he should take home?

Input

The first line of input will contain a single integer $n, 1 \leq n \leq 2 \cdot 10^5$, denoting the number of turtles he saw in a line at the lake.

The next line of input will contain n space-separated integers, denoting the hunger of each turtle $h_i, 1 \leq h_i \leq 10^4$.

The next line of input will contain n space-separated integers, denoting the obedience of each turtle $o_i, 1 \leq o_i \leq 10^4$.

The next and final line of input will contain n space-separated integers, denoting the size of each turtle $s_i, 1 \leq s_i \leq 10^4$.

Output

The first and only line of output should contain two space-separated integers, denoting the index of the turtle in line that he should take home, and its docility. The first turtle in line has index 0, while the last one in line has index $n - 1$.

Sample Input 1

```
5
3 2 5 4 10
9 9 1 11 8
8 5 8 9 10
```

Sample Output 1

```
4 70
```

F – Problem Permutations

Filename: permutations

Determining the order in which the problems appear for HSPC is no small feat, and Nimmi would like to try her hand at it! Since each problem is unique, this means that the number of distinct orderings is simply the factorial of the number of problems. For example, for a problem set containing only five problems, there are 5! = 120 different ways to order that problem set. But Nimmi finds small factorials boring, and she wants to consider problem sets with *a large number of* different problems: potentially *thousands*! Since even these numbers are too big for her to evaluate the factorial, she is instead only concerned with how many *trailing zeroes* the answer has. Trailing zeroes are any zeroes which appear at the *end* of a number's digits. For example, the number 71 has no trailing zeroes, while the number 10,370 has exactly one trailing zero. Your job is to automate this task for Nimmi!

Input

The first and only line of input will contain a single integer $n, 1 \leq n \leq 5 \cdot 10^4$.

Output

The first and only line of output should contain a single integer, denoting the number of trailing zeroes in its factorial.

Sample Input 1

```
12
```

Sample Output 1

```
2
```

Sample Input 2

```
20
```

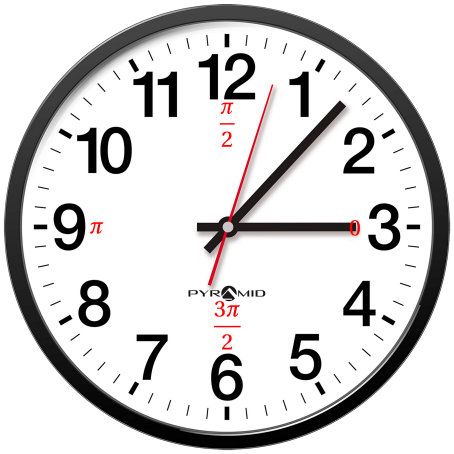
Sample Output 2

```
4
```

G – Temporal Tantrums

Filename: temporal

Dr. Shi has recently become a doctor, a feat which he and his colleagues are most proud of! Following this incredible achievement, however, he has very little free time! He finds himself looking at his clock (which is **analog**!) whenever he is asked if he is free, so much so that he begins wondering about the hour hand of the analog clock. Specifically, he wants to figure out the angle that the hour hand makes on the circular clock with the positive X-axis. With every second that elapses, the hour hand moves *just* a little bit, which results in completely different angles for every time of the day! For a given time of the day, he wishes he could tell his friends what the angle of the hour hand makes as a fun bit of trivia, but alas, his priorities lie elsewhere. He has enlisted your help in figuring out the angle given a time of day, so that he may impress his friends with his niche knowledge of clock hands!



Analogue clock displaying the angles as used in the problem

Input

The first and only line of input contains three integers, with the hours followed by minutes followed by seconds, and each separated by a single colon (`:`), thus giving a valid 12-hour timestamp (hours will be values between 1 and 12).

Output

The first and only line of output should contain a single real number, denoting the angle the hour hand makes with the clock, in radians. Your answer will be considered correct if it differs by no more than 10^{-4} in absolute value from the correct answer.

Sample Input 1

```
6:00:00
```

Sample Output 1

```
4.7123889804
```

Sample Input 2

```
9:00:00
```

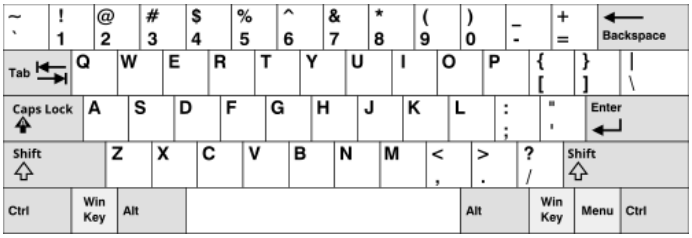
Sample Output 2

```
3.1415926536
```

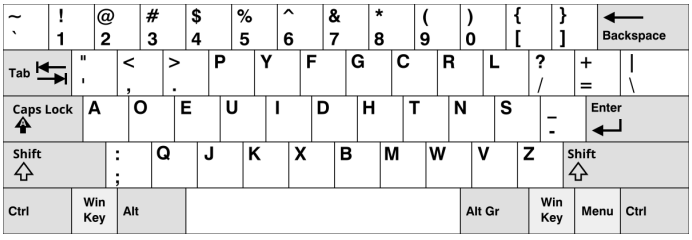
H – Learning the layout

Filename: layout

It's no secret that Gian *loves* speedtyping, and he also happens to be very good at it; he is top 50 and top 100 *worldwide* on Monkeytype and Typeracer, respectively. Against his better judgment, and to further challenge himself, he has decided to purchase a new keyboard with a new layout called the *dvorak* keyboard layout! A keyboard layout is just a specific arrangement of the keys on the keyboard. For example, the one you are typing on right now is probably qwerty (the top row should have keys Q, W, E, R, T, Y, U, I, O, P, ...). Since Gian is still using the muscle memory he's built up over the years with qwerty, what he's typing comes out as complete nonsense on dvorak! Can you translate Gian's messages using the knowledge of how both layouts look?



QWERTY Keyboard Layout



DVORAK Keyboard Layout

Input

The first and only line of input is Gian's message, as it was typed out on the dvorak keyboard. Gian's message will have a length no greater than 10^5. He may press any key in the middle three rows of the dvorak keyboard except tab, caps lock, enter, and shift. He may also press the spacebar.

Output

The first and only line of output should be Gian's message translated, as described earlier.

Sample Input 1

```
d.nnrw ,rpne
```

Sample Output 1

```
hello, world
```

Sample Input 2

```
.qj.nocrp
```

Sample Output 2

```
excelsior
```

I – Musical Mystery

Filename: musical

Yana has been playing a lot of Minecraft recently, and in her newest world, she decided to set some ambitious goals, one of which is to collect all of the game's music discs! Right now, she's focused on collecting the ones that are dropped when a Creeper (a hostile entity in the game) is killed by a Skeleton (another hostile entity in the game). She knows that there are k distinct music discs any of which could be dropped at random from any single Creeper death, each with probability $\frac{1}{k}$. Yana fears collecting them may take a while, but she will only stop collecting once she has at least one copy of every single disc available! She would like you to figure out the *expected number* of Creepers that need to be killed by Skeletons before she collects all k music discs.

Input

The first and only line of input is a single integer $k, 1 \leq k \leq 10^5$: the number of music discs that can be collected in the game as described above.

Output

The first and only line of output is a real number, denoting the expected number of creepers that need to be defeated. Your answer will be correct if it is within 10^{-3} of the correct answer.

Sample Input 1

```
12
```

Sample Output 1

```
37.23852813852814
```

J – Making a Username

Filename: username

Samarth is a budding undergraduate researcher in Computer Science, and as such, he absolutely needs an arXiv account! However, he has been spending many hours thinking about what his username should be. He has some conditions for his username:

1. His username will depend on two values, x and k.
2. The username will only use English lowercase letters; specifically, only the first k letters.
3. For any pair of indices i and j, if $i < j$ and $j - i$ is a multiple of x, then username$_i$ and username$_j$ must not be equal. In other words, if $|j - i|$ is a multiple of x then the i-th and j-th letters of the username must be different from each other.

Given these constraints, Samarth would like to know the length of the smallest string(s) for which a username with the constraints above does not exist.

Input

The first and only line of input contains two integers, $1 \leq x \leq 500$ and $1 \leq k \leq 26$, each separated by a single space.

Output

The first and only line of output should contain a single integer, denoting the answer as described above.

Sample Input 1

```
2 1
```

Sample Output 1

```
3
```

Sample Input 2

```
3 2
```

Sample Output 2

```
7
```

Sample Input 3

```
5 4
```

Sample Output 3

```
21
```

K – Timekeeping

Filename: timekeeping

Parshatd is very busy studying for his PhD qualifiers, but has a very keen interest in hourglasses. The hourglasses he has are capable of measuring s minutes. Since he *loves* to fiddle with his hourglass, he decides he will flip it every k minutes; that is, he will let the hourglass run for *exactly* k minutes – it does not matter to him if there is sand still falling, or if the sand has already finished falling, he will still let it be for k minutes – and then instantaneously flip the hourglass (on account of his lightning-fast reflexes!). After m minutes of doing this (not necessarily a multiple of k), he will leave for a meeting with Dr. Giri. Note that if m is a proper multiple of k, then he would flip the hourglass before leaving for his meeting. When he gets back to his office after the meeting, he would like to know for how many minutes *after he left* the sand continued to fall.

Input

The first and only line of input contains exactly three integers, each separated by a single space: s, k and m, $1 \leq s, k, m \leq 10^9$.

Output

The first and only line of output should contain a single integer, the number of minutes for which sand continued to fall after Parshatd left for his meeting.

Sample Input 1

```
8 8 12
```

Sample Output 1

```
4
```

Sample Input 2

```
5 10 17
```

Sample Output 2

```
0
```

Sample Input 3

```
16 7 7
```

Sample Output 3

```
7
```

L – Area Equality

Filename: area

Azam has discovered a fun new game she enjoys playing in her free time! She has a geometric plane, on which there are n axis-aligned squares (all sides of the squares are parallel to the x or y axes) of varying sizes. These squares do not overlap, as Azam does not like overlapping polygons. She wants to place a horizontal line down somewhere on this plane (i.e., a line of the form $y = k$, for some value k) so that the sum of the areas of all squares above the line and the sum of the areas of all squares below the line are equal; if such a line cuts through a square, then the part of this square below the line is included in the area for the lower section, and the part of the square above the line is included in the area for the upper section. However, to spice this game up a little bit, since there may be several lines that satisfy the above requirement, she would like you to find the *lowest* horizontal line (i.e., the line that minimizes k).

Input

The first line of input contains a single integer $1 \leq n \leq 5 \cdot 10^4$, the number of squares on Azam's plane. This is followed by n lines of input. The i-th line after the first line contains three integers, each separated by a single space: x_i, y_i, l_i, where $0 \leq x_i, y_i, l_i \leq 10^9$, (x_i, y_i) gives the coordinates of the bottom-left corner of the i-th square, and l_i is its length.

Output

The first and only line of output should contain a single real number, denoting the smallest k that divides the squares equally. Your answer will be judged as correct if it is within 10^{-4} of the correct answer.

Sample Input 1

```
2
0 0 1
2 2 1
```

Sample Output 1

```
1.000000000
```