NCPC 2013

5th October 2013

International Collegiate Programming Contest

event sponsor

Problems

A Planting Trees
B Boiling Vegetables
C Number Trick
D Robert Hood
E Virus Replication
F Timebomb
G Erase Securely
H Pinball
I Dance Reconstruction
J Dartboard
K Cliff Walk

Do not open before the contest has started.

Advice, hints, and general information

- Your submissions will be run multiple times, on several different input files. If your submission is incorrect, the error message you get will be the error exhibited on the first input file on which you failed. E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either "wrong answer" or "run time error", depending on which is discovered first.
- For problems with floating point output, we only require that your output is correct up to either a relative or absolute error of 10^{-6}. For example, this means that
  - If the correct answer is 0.05, any answer between 0.049999 and $.050001$ will be accepted.
  - If the correct answer is 50, any answer between 49.99995 and 50.00005 will be accepted.

  Any reasonable format for floating point numbers is acceptable. For instance, "17.000000", "0.17e2", and "17" are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

Problem A
Planting Trees
Problem ID: trees

Photo from Wikipedia, after U. S. Department of Agriculture

Farmer Jon has recently bought n tree seedlings that he wants to plant in his yard. It takes 1 day for Jon to plant a seedling[1], and for each tree Jon knows exactly in how many days after planting it grows to full maturity. Jon would also like to throw a party for his farmer friends, but in order to impress them he would like to organize the party only after all the trees have grown. More precisely, the party can be organized at earliest on the next day after the last tree has grown up.

Help Jon to find out when is the earliest day when the party can take place. Jon can choose the order of planting the trees as he likes, so he wants to plant the trees in such a way that the party will be as soon as possible.

Input

The input consists of two lines. The first line contains a single integer N ($1 \leq N \leq 100\,000$) denoting the number of seedlings. Then a line with N integers t_i follows ($1 \leq t_i \leq 1\,000\,000$), where t_i denotes the number of days it takes for the ith tree to grow.

Output

You program should output exactly one line containing one integer, denoting the earliest day when the party can be organized. The days are numbered $1, 2, 3, \ldots$ beginning from the current moment.

Sample Input 1	Sample Output 1
4 2 3 4 3	7

Sample Input 2	Sample Output 2
6 39 38 9 35 39 20	42

[1]Jon isn't particularly hardworking.

This page is intentionally left (almost) blank.

Problem B
Boiling Vegetables
Problem ID: vegetables

Photo by Martin Cathrae

The trick to boiling vegetables is to make sure all pieces are about the same size. If they are not, the small ones get too soft or the large ones get undercooked (or both). Fortunately, you have heard of the kitchen knife, but your parents' warnings of using sharp instruments still echoes in your head. Therefore you better use it as little as possible. You can take a piece of a vegetable of weight w and cut it arbitrarily in two pieces of weight w_{left} and w_{right}, where $w_{\text{left}} + w_{\text{right}} = w$. This operation constitutes a "cut". Given a set of pieces of vegetables, determine the minimum number of cuts needed to make the ratio between the smallest and the largest resulting piece go above a given threshold.

Input

The input starts with a floating point number T with 2 decimal digits, $0.5 < T < 1$, and a positive integer $N \leq 1\,000$. Next follow N positive integer weights $w_1, w_2, ..., w_N$. All weights are less than 10^6.

Output

Output the minimum number of cuts needed to make the ratio between the resulting minimum weight piece and the resulting maximum weight piece be above T. You may assume that the number of cuts needed is less than 500.

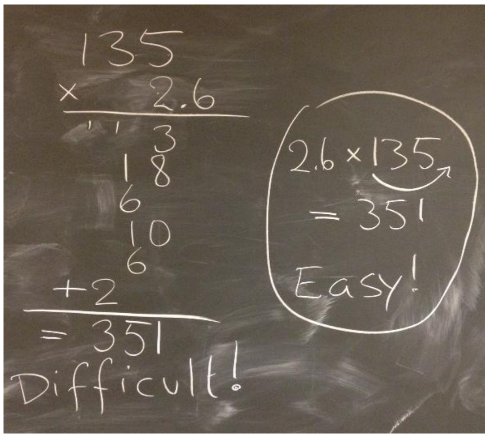
To avoid issues with floating point numbers, you can assume that the optimal answer for ratio T is the same as for ratio $T + 0.0001$.

Sample Input 1	Sample Output 1
0.99 3 2000 3000 4000	6

Sample Input 2	Sample Output 2
0.80 2 1000 1400	3

This page is intentionally left (almost) blank.

Problem C
Number Trick
Problem ID: trick



Lukas is to hold a presentation on useful mathematical tricks. E.g., to take the square root of a number you just need to remove the first half of the number. To convince his audience he uses the well tested method of proof by example: $\sqrt{25} = 5$ and $\sqrt{5776} = 76$ so the method obviously works. To multiply a number by $X = 2.6$ all you have to do is move the first digit to the end of the number, $135 \times 2.6 = 351$ and $270270 \times 2.6 = 702702$.

Lukas wants to demonstrate that this last method works for any X. To do this he will ask his audience for values of X and then show them example multiplications for which the method works. Lukas has noticed that he can not just pick arbitrary numbers for his examples, so now he wants your help. Can you write a program that given X gives a list of integers for which multiplying by X is equivalent to moving the first digit to the end of the number? Lukas does not like very large numbers so do not list any numbers with more than 8 digits.

Input

The input is a single decimal number X ($1 \leq X < 1000$) with at most 4 digits after the decimal point.

Output

Output a list of all positive integers less than 10^8 for which Lukas' second trick works. Write the numbers in ascending order, one number per line. If the list is empty, output instead "`No solution`".

Sample Input 1	Sample Output 1

`2.6`	`135`
	`270`
	`135135`
	`270270`

Sample Input 2	Sample Output 2
`3.1416`	`No solution`

This page is intentionally left (almost) blank.

Problem D
Robert Hood
Problem ID: roberthood

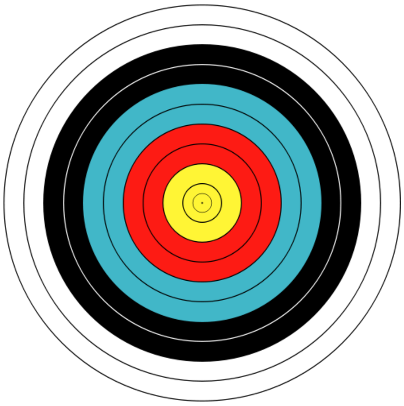


Illustration by Alberto Barbati, released under CC BY-SA.

Robert Hood, a less famous sibling of the Robin Hood, is fed up. Despite him being a young, talented archer he never seems to reach quite the same level as his legendary brother, and so he decided to come up with rules for a new archery contest, in which he will stand a better chance of winning.

The rules for the new kind of archery contest are quite simple: the winner is no longer the one who can score the most points, but instead the one who can achieve the longest distance between any pair of arrows hitting the target. Your task is to write the code to calculate that distance.

A contestant is allowed a number of arrow shots, and the coordinates of the arrows successfully hitting the target are given as a list of pairs. The coordinate system is Cartesian with the origin in the centre of the archery butt. If a contestant does not hit the target with at least two arrows he or she is disqualified and removed from the input data.

Your task is to calculate the score for one contestant that has not been disqualified.

Input

The input starts with a line containing a single positive integer $C, 2 \leq C \leq 100\,000$, representing the number of shots successfully hitting the target for this particular contestant. Each following line contains a pair of integer coordinates separated by a space, representing the x- and y-coordinates of a successful shot. The absolute value of any coordinate does not exceed $1\,000$.

Output

Print the longest distance between any pair of arrows as a floating point number on a single line. The answer is considered correct if it has a relative or absolute error of less than 10^{-6}.

Sample Input 1	Sample Output 1
2 2 2 -1 -2	5.0

Sample Input 2	Sample Output 2
5 -4 1 -100 0 0 4 2 -3 2 300	316.86590223

This page is intentionally left (almost) blank.

Problem E
Virus Replication
Problem ID: virus

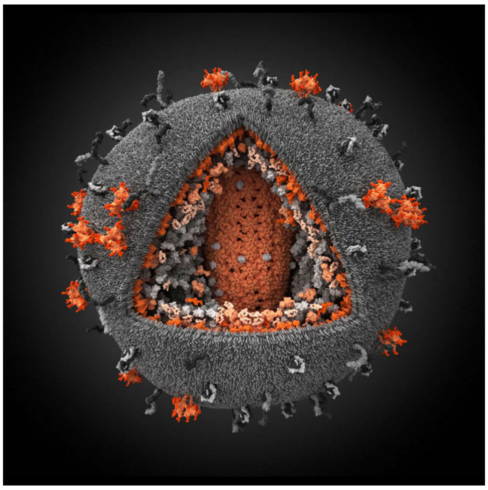


Image from Microbe World

Some viruses replicate by replacing a piece of DNA in a living cell with a piece of DNA that the virus carries with it. This makes the cell start to produce viruses identical to the original one that infected the cell. A group of biologists is interested in knowing how much DNA a certain virus inserts into the host genome. To find this out they have sequenced the full genome of a healthy cell as well as that of an identical cell infected by a virus.

The genome turned out to be pretty big, so now they need your help in the data processing step. Given the DNA sequence before and after the virus infection, determine the length of the smallest single, consecutive piece of DNA that can have been inserted into the first sequence to turn it into the second one. A single, consecutive piece of DNA might also have been removed from the same position in the sequence as DNA was inserted. Small changes in the DNA can have large effects, so the virus might insert only a few letters, or even nothing at all.

Input

The input consists of two lines containing the DNA sequence before and after virus infection respectively. A DNA sequence is given as a string containing between 1 and 10^5 upper-case letters from the alphabet $\{$A, G, C, T$\}$.

Output

Output one integer, the minimum length of DNA inserted by the virus.

Sample Input 1	Sample Output 1
`AAAAA` `AGCGAA`	`3`

Sample Input 2	Sample Output 2
`GTTTGACACACATT` `GTTTGACCACAT`	`4`

This page is intentionally left (almost) blank.

Problem F
Timebomb
Problem ID: timebomb

Photo by Chris Welch

You and your teammates from the anti-bomb squad of the local police have been called to defuse a bomb found in the only pub in town. Fearing the tragic consequences this might produce, you go to the scene as quickly as possible. After some research, you learn that the bad guys have created a tricky way to allow them to defuse the bomb at will. You find a remote control with a button that you can take to a safe place. You also find that it is possible to connect to the bomb through a wireless connection and retrieve an ASCII representation of a code every 2 seconds. The bomb then gets defused if the button is pressed when the code is a number divisible by 6. But you have to be careful. If you press the button when the ASCII representation of the code is not a number divisible by 6 or has an invalid representation for any digit, the bomb will explode instead. You have to rely on your programming skills to write a program able to tell you if it is safe to press the button, before it blows out the pub (and the beer).

Input

The input consists of an ASCII representation of a code. This code has between 2 and 8 digits. Each digit is represented by 5 rows and 3 columns of characters, which can be either a space or a star character '*'. No other type of character (except for the new line character) will ever appear in the input. There is also one column of spaces (and only spaces) to separate each digit. After the last digit you will find a column of new line characters. Note that although every digit will always be of size 5×3, there is no guarantee it will represent a valid digit between 0 and 9 inclusive. The valid 5×3 representations for each digit are given below in Figure F.1.

```
### ### ### ### ### ### ### ### ### ###
***   * *** *** * * *** *** *** *** ***
* *   *   *   * * * *   *     * * * * *
* *   * *** *** *** *** ***   * *** ***
* *   * *     *   *   * * *   * * *   *
***   * *** ***   * *** ***   * *** ***
```

Figure F.1: The hash '#' characters on the top are there only to mark the 3 columns used for a digit and are not part of the digits' representation.

The code can have leading zeros, hence an ASCII representation of, for example, 00000076 represents the number 76. You may also safely assume that every valid code will correspond to a strictly positive number.

Output

Print one line with “`BEER!!`” if it is safe to press the button and defuse the bomb, and “`BOOM!!`” otherwise.

Sample Input 1

```
***   * * * *** *** *** ***
* *   * * *   *   *   * *
* *   * *** *** *** *** ***
* *   *   * *     * *   * *
***   *   * *** *** *** ***
```

Sample Output 1

```
BEER!!
```

Sample Input 2

```
  *   * *** *** *** * *
  *   * **    * * * * *
  *   * *** *** *** ***
  *   * *   *   * *   *
  *   * *** *** ***   *
```

Sample Output 2

```
BOOM!!
```

Sample Input 3

```
*** ***   * *** ***   *
*   * *   * * *   *   *
*** * *   * *** ***   *
  * * *   *   * *     *
*** ***   * *** ***   *
```

Sample Output 3

```
BOOM!!
```

Sample Input 4

```
*** *** *** * * ***
  *   *   * * * * *
***   * *** *** ***
*     *   *   * * *
***   * ***   * ***
```

Sample Output 4

```
BEER!!
```

Problem G
Erase Securely
Problem ID: erase

Photo by Hometown Beauty

Jon Marius is the computer expert at his company and has now been tasked to find some software for erasing data properly. It is very important that the data should not be recoverable afterwards, so it should be overwritten on the hard drive several times. Unable to find any free program up to the task, Jon Marius decides to write such a program himself. The user interface is simple, it only asks for the file to be destroyed and n, the number of times it should be overwritten. This number can range from 1 (quick deletion) to 20 (maximum security). Jon Marius processes the file bit by bit and does not consider writing a zero where there was already a zero as really overwriting. So for each of the n sweeps, he overwrites each zero with a one and each one with a zero.

Jon Marius knows that independent testing is important, so he has asked you to write the verification routine. He will not listen to your objections to the algorithm so eventually you give in.

Input

The first line of the input contains a single integer $1 \leq N \leq 20$. The two following lines each contain a string containing only the characters `0` and `1`. The first of these lines represent the bits of the file before deletion and the second the bits on the same position on the hard drive after the file has been deleted. The length of these strings are the same and between 1 and 1 000 characters.

Output

Output a single line containing either the words "`Deletion succeeded`" if each bit is switched N times or "`Deletion failed`" if this is not the case.

Sample Input 1	Sample Output 1
1 10001110101000001111010100001110 01110001010111110000101011110001	Deletion succeeded

Sample Input 2	Sample Output 2
20 0001100011001010 0001000011000100	Deletion failed

This page is intentionally left (almost) blank.

Problem H
Pinball
Problem ID: pinball

Image from Florence Ivy

Maria is quite addicted to pinball. She can shoot the ball to any position at the top of the board, but she cannot predict where the ball will end when it falls down, because it hits many bumpers on its way down.

She decided to model the pinball table as line segments and assume that the ball is a point that falls from infinite height. The ball falls straight vertically unless there is a segment immediately below it, in which case it follows the direction of the segment downwards until its end.

As you would expect the segments are closed, that is an endpoint is part of its segment. Pairs of segments do not intersect, not even at endpoints, and none is vertical or horizontal. Segments are not given in any specific order.

Input

The first line contains an integer N ($0 \leq N \leq 100\,000$), the number of segments. Then N lines follow, each with four integers x_1 y_1 x_2 y_2, the coordinates of a segment ($-1\,000\,000 \leq x_i, y_i \leq 1\,000\,000$). The last line contains an integer x_0 ($-1\,000\,000 \leq x_0 \leq 1\,000\,000$), the initial x-coordinate of the ball.

Output

Output a single integer x_T, the final x-coordinate of the ball.

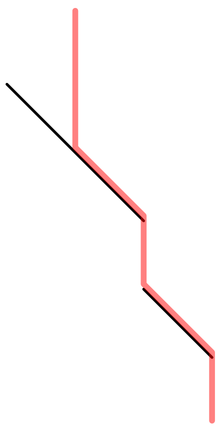


Figure H.1: Sample input 1

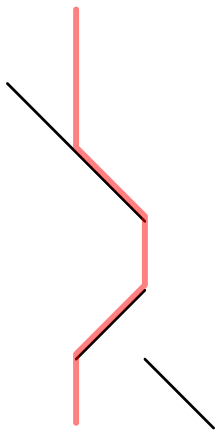


Figure H.2: Sample input 2

Sample Input 1	Sample Output 1
2 -1 1 1 -1 1 -2 2 -3 0	2

Sample Input 2	Sample Output 2
3 -1 1 1 -1 1 -2 0 -3 1 -3 2 -4 0	0

Problem I
Dance Reconstruction
Problem ID: dance

Photo by Richard Taylor

Marek loves dancing and he has danced a lot in the last couple of years. He has actually danced so much that he became too good in all of the traditional dances like swing, salsa, ballroom and hip-hop and now all partners he dances with can not keep up with him. Therefore he started to invent his own dances and even tries to convince other people to dance these new dances with him.

Marek got really excited when he heard about the coming wedding of his best friend Miroslav. For a whole month he worked on a special dance for the wedding. The dance was performed by N people and there were N marks on the floor. There was an arrow from each mark to another mark and every mark had exactly one incoming arrow. The arrow could be also pointing back to the same mark.

At the wedding, every person first picked a mark on the floor and no 2 persons picked the same one. Then Marek played some music and every 10 seconds there was a loud signal when all dancers had to move along the arrow on the floor to another mark. The placement of the marks was such that everybody could follow the arrow to the next mark in 10 seconds without any trouble. If an arrow was pointing back to the same mark, the person at the mark just stayed there and maybe did some improvized dance moves on the spot.

A year has passed since Miroslav's wedding and another wedding is coming up. Marek would like to do a similar dance at this wedding as well. He lost all the drawings he had, but luckily he found two photos from exactly when the dance started and when it ended. Marek also remembers that the signal was triggered K times during the time the song was played, so people moved K times along the arrows.

Given the two photos, can you help Marek reconstruct the arrows on the floor? On the two photos it can be seen for every person to which position he or she moved. Marek therefore numbered the people in the first photo from 1 to N and then wrote the number of the person whose place they took in the second photo.

Marek's time is running out, so he is interested in any placement of arrows that could produce the two photos.

Input

The first line of the input contains two integers N and K, $2 \leq N \leq 10\,000, 1 \leq K \leq 10^9$. The second line of the input contains N space separated integers $a_1, \ldots, a_N$, denoting that dancer number i ended at the place of dancer number a_i. You may assume that $1 \leq a_i \leq N$ for all i and every number between 1 and N inclusive appears exactly once in the sequence.

Output

If it is impossible to find a placement of arrows such that the dance performed K times would produce the two photos, print "`Impossible`". Otherwise print N numbers on a line, the i-th number denoting to which person the arrow leads from person number i.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

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

Sample Output 2

```
2 3 4 1
```

Problem J
Dartboard
Problem ID: dartboard

Jaap is playing darts at the local pub with a group of friends. His darts throwing skills are not that great, so he just tries to aim at the center of the dartboard. His mathematical skills are better though, and he wonders what is his expected score for one dart.

After a while Jaap estimates that his darts hit the dartboard (or often miss it) with a probability distribution that depends only on the radius r from the center of the board, and has the Gaussian form[2]

$$f(r) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}.$$

That is, the probability of hitting a small surface area $\Delta x \cdot \Delta y$ at a distance r from the center is given by $f(r)\Delta x \cdot \Delta y$. Here σ denotes the standard deviation, and Jaap found out that this depends strongly on how many beers he has had.

For those not familiar with the game of darts, a dartboard is depicted below. The score for hitting each of the regions of the dartboard is as follows:

- the inner *bull's eye* is worth 50 points;
- the *bull* annulus is 25 points;
- each pie has worth of the respective number 1 up to 20, but
- the inner *triple ring* has triple the worth of the pie, while
- the outer *double ring* has double the worth.

Finally, if the dart lands outside the double ring, the score is zero. Note that the pies of all numbers have equal area.

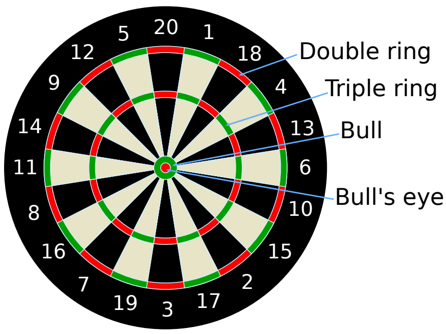


Figure J.1: A standard dartboard (from Wikimedia, CC BY-SA 3.0 licensed by Tijmen Stam).

Input

The first line contains 6 floating point numbers of strictly increasing size: the radii of the *bull's eye*, *bull*, inner and outer *triple ring*, and inner and outer *double ring*, all in centimeters. The second line contains the standard deviation σ in centimeters as a floating point number. All floating point numbers are in the range $[10^{-3}, 100]$.

[2]This implies that the total probability is never larger than one.

Output

Print the expected score of one dart for Jaap as a floating point number on a single line. The answer should be correct up to either a relative or absolute error of 10^{-4}.

Sample Input 1	Sample Output 1
`1.27 3.1 10.9 11.7 16.2 17.0` `17.0`	`5.266210658`

Sample Input 2	Sample Output 2
`1.27 3.1 10.9 11.7 16.2 17.0` `0.5`	`49.00690019`

Sample Input 3	Sample Output 3
`0.1 0.2 0.3 0.4 99.9 100` `20`	`10.50283655`

Problem K
Cliff Walk
Problem ID: cliffwalk

Image from Nuala

One morning last summer, Charlotte was watching the moon and the sun and observed that the moon was full. As she lives along the Atlantic coast she knows that this means a larger variation in the tide compared to first and last quarter. With no rain in the air, it seemed like a perfect week for walks at the beach by the cliffs.

The tide is dangerous when walking at the beach between the sea and the cliff wall. As the water rises, you may get trapped. Therefore it is important to plan the walk according to the behaviour of the tide.

One simple way of cliff walk planning is just to start walking and turn around at low tide. The problem is that on a rocky beach, you want the rocks to dry for one hour before entering them. It could therefore actually be safe to continue the walk a bit further even after low tide. Note that the beach is mostly made of sand and the rocks have many cracks in them, so we assume that all areas are flooded or drained at the exact moment when the tide reaches their height, irrespective of the heights of the neighbouring areas.

The beach has been surveyed and a map is available where each 10×10m square has a certain height. Each square can only be entered from the four neighbouring squares to the north, south, east and west. It is only possible to pass between two squares of height z_1, z_2 if the absolute height difference $|z_1 - z_2|$ is at most 1 meter. Charlotte walks in such a way that it takes a constant amount of time to pass from one square to another and during the whole time period both squares must be dry. Charlotte can also decide to stay at a square for any amount of time.

The tide behaves differently at different places on the Earth depending on the sea bottom, coast line etc. Charlotte knows that it is possible to approximate the tide's water level v in meters as $v = 0.5a \cdot (\cos(t\frac{2\pi}{12}) + 1)$, where t is time in hours since the last high tide and a is height in meters depending on the location, time of the year, etc.

Charlotte will start and finish her walk at her home. She limits her time away from home to only one tide interval, so you may assume that $0.0 \le t \le 12.0$. How far from home is she able to get and still return safely back?

Input

The first line of the input contains two floating point numbers a, $0.0 < a < 15.0$, and m, $0.1 \le m \le 60.0$, the number of seconds it takes to pass one square on the map. The second line contains four integers W, H, X and Y where $1 \le W, H \le 200$, $0 \le X < W$ and $0 \le Y < H$. W and H are the width and height of the map of the coast, X and Y describes the coordinate (X, Y) of Charlotte's home.

Then follow H lines each containing W space separated integers, describing the height in millimetres of each 10×10m surveyed square compared to extreme low tide. You can assume that the height of each square will be at least 0 and at most $20\,000$ milimetres. The first number on the first line corresponds to coordinate $(0, 0)$. Charlotte's home will always be dry.

Output

Output one line with the maximum Euclidean distance that Charlotte can get from home. The distance between two squares should be measured between their centers. The answer is considered correct if it has an absolute or relative error of at most 10^{-6}.

To avoid problems with floating point numbers, the result is guaranteed to be the same for all walking speeds m' where $0.999m < m' < 1.001m$.

Sample Input 1	Sample Output 1
2.0 10.0 3 3 0 0 2001 1000 100 1001 10000 200 100 0 0	20

Sample Input 2	Sample Output 2
4.0 30.0 6 2 2 0 73 1001 4001 1001 76 70 70 2001 3001 2001 72 71	22.36067977