培正喇沙編程挑戰賽
LA SALLE – PUI CHING PROGRAMMING CHALLENGE

4TH · 2019

MAY 13, 2019 (MONDAY)

LA SALLE COLLEGE

ID	NAME	TIME LIMIT	MEMORY LIMIT
A	A Billionaire	0.5 seconds	256 MB
B	Blokus Duo	1 second	256 MB
C	camelCaseCounting	1 second	1024 MB
D	Drawing Circles	1 second	256 MB
E	Eat More	0.5 seconds	256 MB
F	Find the Base	1.5 seconds	256 MB
G	Guessing Game	0.5 seconds	256 MB
H	Hacking	0.5 seconds	256 MB
I	Ice-cream Sampler	1 second	256 MB
J	Just A $10 Note	0.5 seconds	256 MB
K	Kth number in Byteland	1 second	256 MB
L	LRTB and TBRL	1 second	256 MB

Pascal	`ppcx64-3.0.0 -O2 -Sg -v0 -dEVAL -XS`
C	`gcc-5.4.0 -static -Wno-unused-result -DEVAL -lm -s -O2`
C++	`g++-5.4.0 -static -std=c++11 -Wno-unused-result -DEVAL -lm -s -O2`

Compilation Time Limit: 10 seconds   Memory Limit: 512 MB

AUTHORS

Chow Kwan Ting Jeremy, Poon Lik Hang,

Wong Man Hang, Wong Tsz Chun, Wong Yik Chun

NOTES

1. Output strings are checked case-insensitively.
2. Trailing spaces in output lines are allowed.
3. The end-of-line character at the end of the output is optional.
4. Using scientific notation for real number output is not allowed.
5. 64-bit integer types (e.g. `int64` for Pascal, `long long` for C/C++) may be required for some problems. The C format string token is `%lld`.

A Billionaire

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

I wanna be a Billionaire so bad, Buy all of the things I never had.

Billionaire Travie McCoy ft. Bruno Mars

Alice is a big fan of Burno Mars. After she listened to Burno Mars' song "Billionaire", she also wants to buy a lot of things she wants.

Despite wanna buy a lot of luxuries, Alice is not a billionaire at this moment. Currently, Alice just has K dollars. Luckily, she is a genius in investment, she can earn exactly E dollars per day! She plans to use her income and savings to buy the luxuries she wants starting from tomorrow (day 1).

Alice finds that there are N luxury items she wants to buy. The i-th item is priced at c_i dollars. Undoubtedly, Alice is going to buy all of them! However, she also needs to limit her expenditure every day. Therefore, Alice decides not to buy more than one item on a single day. To buy the item i on some day, Alice needs to have at least c_i dollars on that day. In other words, her wealth is not allowed to be negative at any moment. She can also choose not to buy on some days such that she can accumulate her money earned and buy something more expensive later. Now, Alice is curious to know what is the minimum number of days such that she can obtain all N luxury items. Note that Alice is not necessary to buy the luxury items in order.

Input

The first line contains 3 integers, N, K, E, representing the number of luxury items Alice wants to buy, the money she has at the beginning and the money she earns per day.

The second line contains N integers, c_i, representing the price of the i-th item.

It is guaranteed that $1 \leq N \leq 200$ and $1 \leq K, E, c_i \leq 10^7$

Output

Output an integer in a single line, which is the minimum number of days such that Alice can obtain all N items.

Examples

input

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

output

```
5
```

input

```
2 10 10
5 5
```

output

```
2
```

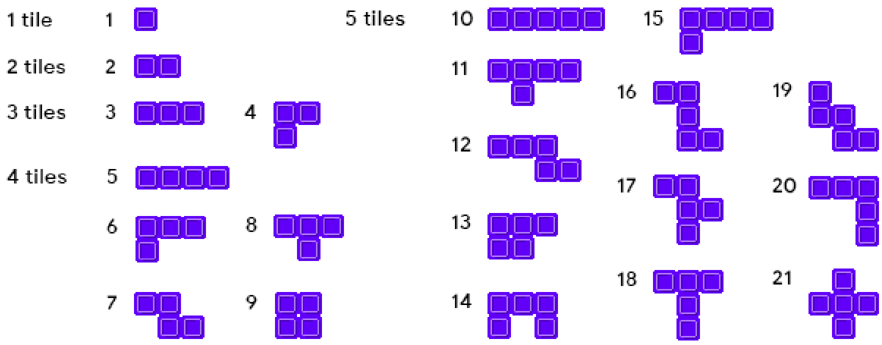
Note

In sample 1, Alice may buy the 2-nd item, which is 5 dollars, on day 2 as she will have $2+2\times 2=6$ dollars on that day. She will have 1 dollar remaining after buying it. Then Alice may buy the 3-rd item on day 3 and the 1-st item on day 5.

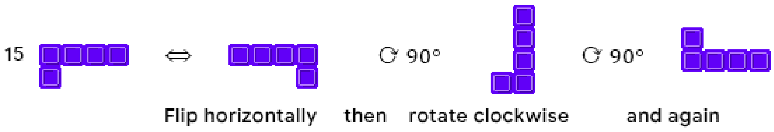
B Blokus Duo

Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output

Blokus Duo is a 2-player version of the Blokus board game. The game is played on a 14×14 grid. The rows are numbered from 1 to 14 from top to bottom. The columns are numbered from 1 to 14 from left to right. The game uses two (purple and orange) sets of 21 different pieces.



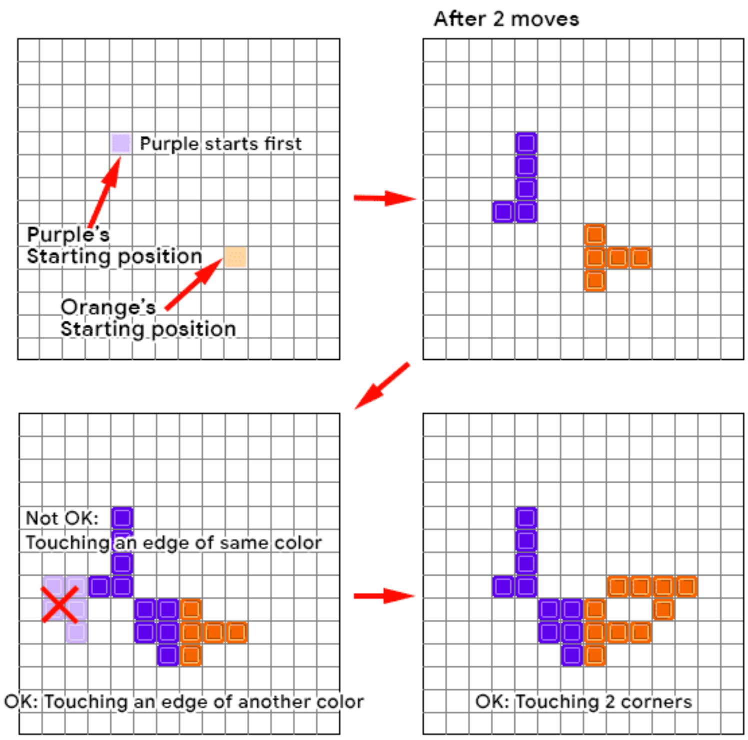
Pieces can be flipped and/or rotated to produce different shapes, as illustrated below:



The two players take turn to place pieces. The player that has the purple pieces goes first. First, he/she must place a piece such that it covers the cell $(5, 5)$. The other (orange) player goes next. He/she must place a piece on empty cells such that it covers the cell $(10, 10)$.

After that, each piece must be placed such that:

- It is placed on entirely empty cells.
- It touches at least 1 corner of 1 or more existing piece(s) of the same color.
- It does not touch any edge of any existing piece of the same color.



If a player cannot make any valid move, he/she must surrender so that the other player may continue placing his/her remaining pieces until he/she cannot do so. The game ends when both players are out of pieces or surrenders. Note that once a player surrenders, even if by mistake (when there existed valid move(s) but the player thought that there wasn't any), he/she cannot make any more moves.

You are doing a research about the strategies so you collected some end game states (the state of the grid). Can you write a program to generate a valid sequence of moves to recreate the end game state?

Input

The input consists of a 14×14 grid that consists of `.` (for empty cell), `P` for a cell that is covered by a Purple piece, or `O` for a cell that is covered by an Orange piece. It is guaranteed that it is a valid end game state and both players placed at least 1 piece. Which means that all rules described above are strictly followed, but it's not guaranteed that no player can make no more moves, because both players could have surrendered early by mistake.

Output

Output any sequence of moves that recreates the given end game state, starting from the empty grid. On the first line output an integer N – the number of moves. The next N lines describe the moves in order. A move should consist of the following the information: `color num_cells row_1 column_1 [row_2 column_2]...`

- `color` is the color of the piece. It should be either `P` (for Purple) or `O` (for Orange)
- `num_cells` is an integer between 1 and 5 inclusive that indicates the size of the piece. (number of cells that it covers)
- After that, output the row and column number of the covered cells in any order using `num_cells` pairs of integers between 1 and 14.

Example

input

```
..............
..............
......PPP.....
.....P.....O..
..PPPP..OO.OO.
......PO....O.
.....PPO..OO..
.....POOO.OO..
.....P...O..OO
......PPPO..O.
........PO....
........P..P..
.........PPP..
..............
```

output

```
11
P 5 5 5 5 6 4 6 5 4 5 3
O 3 10 10 11 10 9 10
P 3 3 7 3 8 3 9
O 4 8 11 7 11 7 12 8 12
P 5 6 7 7 7 7 6 8 6 9 6
O 3 9 13 10 13 9 14
P 5 10 7 10 8 10 9 11 9 12 9
O 4 6 13 5 13 5 12 4 12
P 4 13 10 13 11 13 12 12 12
O 5 8 9 8 8 7 8 6 8 8 7
O 2 5 9 5 10
```

C camelCaseCounting

Time Limit: 1 second
Memory Limit: 1024 MB
Input: standard input
Output: standard output

Camel case is one of the common practices for naming variables in programming, so that phrases, clauses (or even sentences) can be represented as a string with no spaces or punctuation. In camel case, the first letter of each word (except the first word) is written in capital letter. All other letters, including the first letter of the first word, should be written in lower case. For instance, the phrase "camel case" in camel case is `camelCase`, and "to do list" in camel case is `toDoList`. Here are some other examples of valid camel case writing:

- `laSalleCollege` ("la salle college")
- `puiChing` ("pui ching")
- `challenge` ("challenge")
- `oMG` ("o m g")

And here are some examples of invalid camel case writing:

- `ProgrammingChallenge`
- `ILoveCoding`
- `THISproblemISeasy`
- `TLDR`

Given a valid camel case writing S, please find the number of unique substrings that is a valid camel case writing. Here, substring is a contiguous sequence of characters within a string.

Input

The only line contains a string S, which consists of only lowercase letters and uppercase letters. It is guaranteed that S is a valid camel case writing, and $1 \le |S| \le 10^6$ (where $|S|$ is the length of S).

Output

Output a single integer, the number of unique substrings that is a valid camel case writing.

Examples

input	output
`ease`	`9`
`camelCaseCounting`	`130`
`oMG`	`3`

Note

In Sample 1, the 9 unique substrings that are valid camel case writings: `a`, `e`, `s`, `as`, `ea`, `se`, `ase`, `eas`, `ease`.

In Sample 2, several substrings of valid camel case writings: `elCaseCount`, `aseCou`, `cam`, `e`.

D Drawing Circles

Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output

David is a little boy who loves drawing circles on the wall. He is now planning on a N-day project. Before the start of the project, he draws two big circles on the wall, so that they touch each other, and both touch the ground. The one on the left has radius r_1 units, and the one on the right has radius r_2 units.

Then, on each day of the project, David will draw one more circle on the wall, while this circle must fulfill the following requirements:

- It is a circle that does not overlap with any other circles drawn before
- It must touch exactly two other circles drawn before
- It must touch the ground
- Among all possible ways of drawing the new circle according to the rules above, David will draw the one with the largest radius. If they are still more than one possibility, he will draw the leftmost one.

Can you predict the radius of the circle that will be drawn on each of the N days?

Input

The only line contains three integers, r_1, r_2 and N ($1 \le r_1, r_2, N \le 3 \times 10^5$).

Output

Output N numbers, the i-th number denotes the radius of the circle that David will draw on the i-th day.

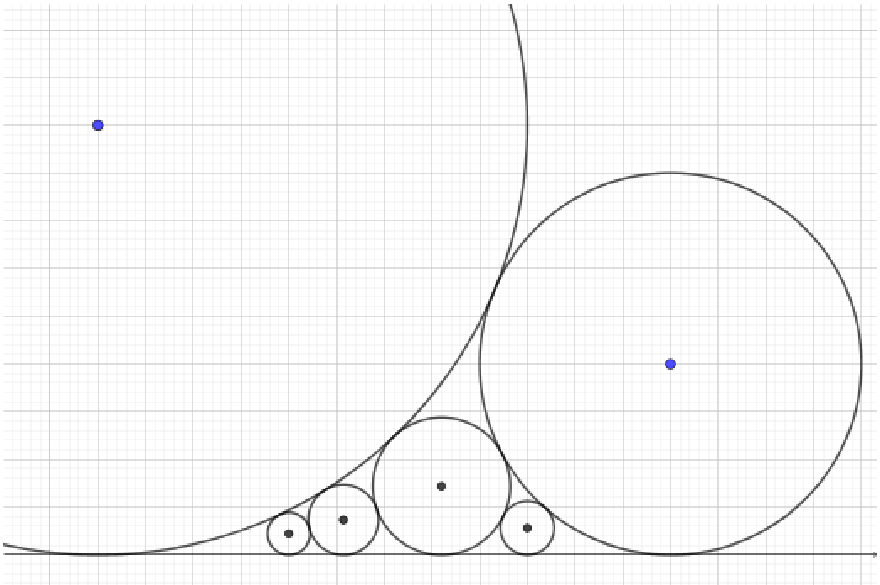
Your answer will be accepted if for each and every radius, the relative error or absolute error, whichever is less, is not greater than 10^{-6}.

Examples

input	output
`9 4 4`	`1.44 0.734694 0.5625 0.444444`
`10 10 1`	`2.5000000000`

Note

In Sample 1, after $N = 4$ days, a part of the wall will look like:



E Eat More

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

HML loves eating!

He has N dishes on his table. Each dish has its own saltiness value, denoted as A_i for the i-th dish.

HML also has a saltiness preference value K. He wants to split the dishes into some groups (possibly one group), and each group forms a consecutive contiguous subsegment of the original dishes. So every dish will be in exactly one group and for the i^{th} group, it contains dishes l_i, $l_i + 1$, ..., r_i. Also, for every group, the absolute value of the sum of saltiness of the dishes belong to the group, should be less than or equal to K. For example, if the group contains dishes l, $l + 1$, ..., r, then $|\sum_{i=l}^{r} A_i| \leq K$, where $|x|$ denotes the absolute value of x.

HML is a curious boy, he wonders how many ways there are to group the dishes. As the number could be large, please output the answer modulo $10^9 + 7$.

Input

The first line consists of two integers, N and K ($1 \leq N \leq 10^5$, $0 \leq K \leq 10^{14}$).

The second line consists of N integers, which is the array A. The i-th integer correspond to A_i ($-10^9 \leq A_i \leq 10^9$).

Output

Output the total number of ways to group the food modulo $10^9 + 7$.

Examples

input	output
4 1 2 -1 -1 0	4
6 2 2 -1 3 -2 0 1	19

Note

In sample 1, there are 4 ways:

1. [2 -1] [-1 0]
2. [2 -1] [-1] [0]
3. [2 -1 -1] [0]
4. [2 -1 -1 0]

This page is intentionally left blank.

F Find the Base

Time Limit: 1.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Today, Bob learned how to write a number in binary representation. Moreover, he learned that he can express a number not only in decimal or binary, but also in octal, in hexadecimal, or even in any bases greater than 1! As Bob is a hardworking student, he wants to revise what he has learned today. Therefore, he would like to express numbers in different bases!

There are N integers written on Bob's notebook and he has a favorite number M. He tries to express all N integers in every base from 2 to B. Suddenly, he realizes that when he writes those N integers in some bases, the digit sum of some integers may become equal to his favorite number M! As M is his favorite number, the more numbers having digit sum equal to M, the more he loves that base. Formally, among all the bases from 2 to B, base x is one of his favorites if, when all N integers are expressed in base x, the number of integers which has a digit sum of M is maximal.

Noted that in this task, in base i, digits of number are represented as {0, 1, 2, ... , i - 1} instead of {0, 1, 2, ... , 9, A, B, C, ... }. So 26 in base 14, i.e. 1(12), has digit sum of $1 + 12 = 13$.

Being a curious person, Bob wants to find out all of his favorite bases. However, he just learned about number systems other than decimal for one day and doesn't familiar with expressing numbers in other representations. So, he wants you to help him to find out all of his favorite bases.

Input

The first line consists of three integers, N, M and B. ($1 \leq N, M \leq 10^5$, $2 \leq B \leq 10^5$.

The second line consists of N integers, which is the numbers written on Bob's notebook. The i^{th} integer correspond to A_i, $1 \leq A_i \leq 10^5$.

Output

The first line consists of one integer, K, the number of Bob's favorite bases.

The second line consists of K integers, all of Bob's favorite bases. Please print the answer in the ascending order.

Examples

input

```
3 3 5
1 3 12
```

output

```
1
4
```

input

```
8 3 15
1 2 2 3 5 16 18 26
```

output

```
3
4 6 14
```

Note

In the first sample, A[] = {1, 3, 30} in base 4, which has 2 number with digit sum of 3, which is maximal among bases from 2 to 5.

G

Guessing Game

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Wukong, Xayah and Yasuo invite you to play a guessing game.

Wukong will first draw a random positive integer W between 1 and 100, including 1 and 100. Each number will be drawn with the same probability, i.e. 1%.

Xayah, Yasuo, and you each have to guess a different number between 1 and 100. A player's goal is to minimize the difference between his/her own number and Wukong's hidden number W. In other words, the one with the number closest to W wins. In case more than one player has the minimum difference, no one wins.

Xayah is the first player to guess, and she guessed X. Yasuo is the next player, and he guessed Y ($X \neq Y$). Now it is your turn, given the numbers X and Y Xayah and Yasuo guessed, you should guess a number Z that is different from X and Y, and also maximize the chance of winning the game.

For example, if the hidden number $W = 50$, and the guesses are $X = 20, Y = 81, Z = 79$. Then you win as Z is the closest number to W (as $|W - Z| = 29$ is smaller than $|W - X| = 30$ and $|W - Y| = 31$). If the hidden number $W = 80$, then no one wins, as $|W - Z|$ and $|W - Y|$ are both equal to 1 which is less than $|W - X| = 60$.

Input

The only line contains two integers X and Y ($1 \leq X, Y \leq 100$, $X \neq Y$).

Output

A single integer Z, which is your guess. If there are more than one possible values that can maximize the chance of winning, please output any one of them.

Example

input	output
20 81	79

This page is intentionally left blank.

H

Hacking

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

The Largest Security Professional Challenge (LSPC) is a team contest. Each team consists of 3 hackers, and team shall hack as many websites as possible in 4 hours.

This year, LSPC's problemset consists of three websites to be hacked, each of different hacking difficulties. The hacking difficulties of the three websites can be represented as integers: d_1, d_2 and d_3, where $d_1 < d_2 < d_3$. This means that website 1 is the easiest to hack, with difficulty of d_1, followed by website 2 with difficulty d_2. Website 3 is the hardest to hack, with difficulty of d_3.

The team from Hackerland consists of three members: Alice, Bob and Charles, each of different hacking skills. The hacking skills of the three members can also be represented as integers: s_A, s_B and s_C, where $s_A < s_B < s_C$. This means that Alice with skill level s_A is the weakest hacker in the team, followed by Bob with skill level s_B. Charles, as the best hacker in the team, has skill level s_C.

A Hacker i is skillful enough to hack website j if and only if the hacker's skill level is **strictly greater than** the hacking difficulty of the website, i.e. $s_i > d_j$.

As the contest time is limited, each hacker can hack **NO MORE THAN ONE** website during the whole contest time.

Given the hacking difficulties of the three websites, and the skill levels of the three hackers, what is the maximum number of websites that can be hacked?

Input

The first line consists of three integers d_1, d_2, d_3 ($1 \le d_1 < d_2 < d_3 \le 100$).

The second line consists of three integers s_A, s_B, s_C ($1 \le s_A < s_B < s_C \le 100$).

Output

A single integer, indicating the maximum number of websites that can be hacked by the team from Hackerland.

Examples

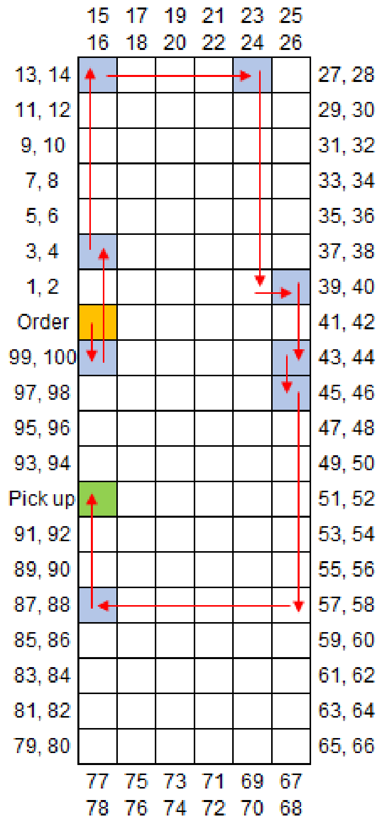
input	output
10 30 50 20 45 100	3
56 78 90 12 34 56	0
31 41 59 26 53 58	2
2 71 82 8 18 28	1
12 22 32 1 10 100	1

Note

Notice that in Sample 5, although Charles is skillful enough to hack any of the websites, he can only hack one of them during the whole contest time.

I Ice-cream Sampler

Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output



Baskin Robbins is a ice-cream parlour brand. Most of the stores offer 31 different flavours, so that customers can try a different flavour every day in a month (note the 31 in the logo). However, the Baskin Robbins Brown in Cheongdam (Seoul, South Korea) offers 100 flavours.

Since there are too many flavours, they sell a ice-cream sampler which let customers try up to 10 flavours.

The flavours, numbered 1 to 100, are located around the counter. After taking your order at the Order point (orange cell), the staff have to scoop the ice-cream by walking to where the flavours are, one after another in any order. When all 10 scoops of ice cream are ready, the sampler is delivered to the Pick Up point (green cell). For some reason, the staff can only walk vertically and horizontally. Note that if two or more chosen flavours are located in the same cell, the staff can scoop all of them without moving. It is allowed to visit a point, including the Order and Pick Up point, more than once.

For example, if your ordered flavours 3, 13, 14, 16, 24, 39, 43, 46, 87, 99, one way that the staff can prepare your order is illustrated with the red arrows. The total walking distance is 37. In fact, this is one of the shortest paths. Given the 10 flavours chosen, can you help the staff minimize the walking distance?

Input

The input consists of 10 integers, between 1 and 100 inclusive, in a single line – the numbers of the 10 chosen flavours given in non-decreasing order. It's allowed for the customers to choose multiple scoops for the same flavour.

Output

Output a single integer – the minimum walking distance needed to scoop all 10 flavours, starting from the Order point and ending at the Pick Up point.

Examples

input

```
3 13 14 16 24 39 43 46 87 99
```

output

```
37
```

input

```
50 50 50 50 50 99 99 100 100 100
```

output

```
15
```

Note

The first sample corresponds to the example in the problem statement.

In the second sample, the customer wants to have 5 scoops of flavour 50, 2 scoops of flavour 99 and 3 scoops of flavour 100.

J Just A $10 Note

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Jane is working as a cashier at a grocery store in Hackerland. This grocery store is now offering a special discount, all the items are on sale, each with a positive integral price of no more than $10.

In the early morning, N customers are queuing outside the store waiting for the store to open, but they have not decided what to purchase yet. When Jane arrives, people in the queue all ask the same question: "I have just a $10 note, do you have enough coins for changes?". What a coincidence! Unfortunately, Jane knows that there are no coins in the store, so she has to ask for some from the bank.

In Hackerland, there are only three types of coins, $5, $2 and $1, and they are of equal weight. Jane wants to minimize the number of coins to carry as the coins are too heavy! Jane predicts that no more customers will be coming. Therefore, she should prepare the least number of coins, so that no matter which item(s) eventually each of the N customers choose to purchase (of course, not more than $10 per person), she has enough amount of coins for changes.

You may assume that when Jane gets back to the store again, every customer has decided what to purchase so she will be able to know the amount of changes needed for every customer, then decide how to distribute the coins for giving the changes. Please note that she must give the exact amount of changes to every customer.

Input

A single integer N, denoting the number of customers queuing outside the store ($1 \leq N \leq 50$).

Output

A single integer, denoting the minimum number of coins Jane carries, such that she has enough amount of coins for changes no matter what item(s) the N customers purchase.

Example

input	output
1	4

Note

There is only one customer, one of the optimal ways is: Jane can prepare four coins: one $5 coin, one $2 coin, two $1 coins. There are ten possible situations:

- The customer purchase items that worth $10 in total, no changes needed!
- The customer purchase items that worth $9 in total, Jane can give one $1 coin back.
- The customer purchase items that worth $8 in total, Jane can give one $2 coin back.
- The customer purchase items that worth $7 in total, Jane can give one $2 coin and one $1 coin back.
- The customer purchase items that worth $6 in total, Jane can give one $2 coin and two $1 coins back.
- The customer purchase items that worth $5 in total, Jane can give one $5 coin back.
- The customer purchase items that worth $4 in total, Jane can give one $5 coin and one $1 coin back.
- The customer purchase items that worth $3 in total, Jane can give one $5 coin and one $2 coin back.
- The customer purchase items that worth $2 in total, Jane can give one $5 coin, one $2 coin and one $1 coin back.
- The customer purchase items that worth $1 in total, Jane can give one $5 coin, one $2 coin and two $1 coins back.

K Kth number in Byteland

Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output

There are thousands of written languages in the world. Some of them are read from left to right, such as English. Some of them are read from right to left, such as Arabic.

In Byteland, programmers come from many different countries. To avoid ambiguity, the mayor of Byteland decided to publish a new written language and also a new number system that remains the same regardless if it is read from left to right or right to left. Moreover, since the mayor loves odd number, he wants the numbers in the new number system consist of odd number of digits. Therefore, the mayor decided that the k^{th} number in the new number system shall be the k^{th} smallest non-negative number in the decimal number system that consists of an odd number of digits and remains unchanged when it is read from either direction.

For example, the 1^{st} number in the new number system is 0, and the 24^{th} number is 232.

As you are a newcomer in Byteland and you are trying to get familiar with the number system, you want to know the k^{th} number in the Byteland!

Input

The input contains only one integer, k ($1 \leq k \leq 10^9$).

Output

Output one integer, the k^{th} number in the new number system in Byteland.

Examples

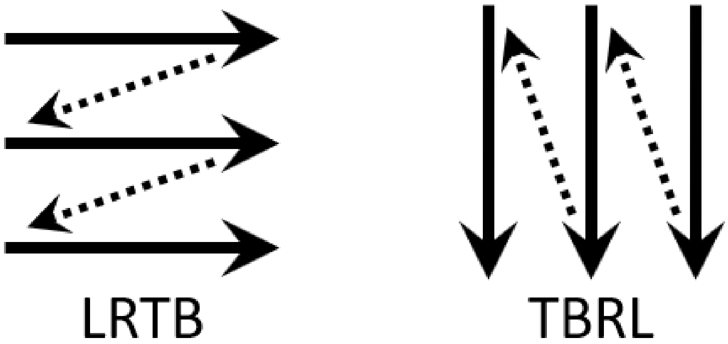
input	output
1	0
24	232

This page is intentionally left blank.

L — LRTB and TBRL

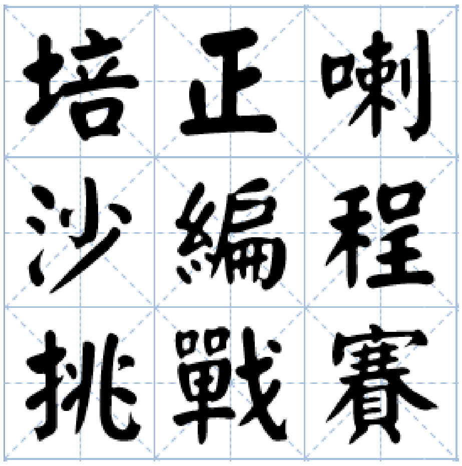
Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output

Chinese is one of the several languages that can be written and read in two directions: "Left-to-right, top-to-bottom" (LRTB), and "Top-to-bottom, right-to-left" (TBRL).



When reading in LRTB, texts are written row by row, and from left to right in each row. Reading starts from the topmost row to the bottommost row.

When reading in TBRL, texts are written column by column, and from top to bottom in each column. Reading starts from the rightmost column to the leftmost column.



For example, when reading the text above, in LRTB, it is "培正喇沙編程挑戰賽". In TBRL, it is "喇程賽正編戰培沙挑".

Lee Sin has learnt N Chinese characters. He has a grid paper with R rows and C columns, and he wants to write one of the N Chinese characters he knows in each of the $R \times C$ cells. To prevent the two reading directions LRTB and TBRL from bringing readers into confusion, Lee Sin plans to write in a way that readers can read the same content no matter they read in LRTB or TBRL.

Please find out the number of ways Lee Sin can fill the grid paper so that readers can read the same content no matter they read in LRTB or TBRL.

Input

The only line contains three integers N, R and C ($1 \leq N \leq 4762$, $1 \leq R, C \leq 1000$).

Output

A single integer, the number of ways Lee Sin can fill the grid paper so that readers can read the same content no matter they read in LRTB or TBRL. As the number may be large, please output it modulo $10^9 + 7$.

Examples

input	output
3 3 3	27

input	output
2 3 2	4

input	output
4 4 5	256

input	output
11 19 31	300916151

Note

In Sample 2, assume the $N = 2$ Chinese characters Lee Sin knows are "李" and "星". The four ways he can fill the grid paper are: ("李李李李李李", "李李星李星星" , "星星李星李李"and "星星星星星星")

In Sample 4, the answer is $61159090448414546291 \mod (10^9 + 7) = 300916151$