Problem B. Rikka with Maximum Subsegment Sum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Maximum Subsegment Sum is a classical problem. When Rikka first saw this problem, she was still an outsider of competitive programming, and now, she has become a problem setter of this grand event.

Therefore, Rikka decides to set a problem about Maximum Subsegment Sum. Given an array x of length m, its maximum subsegment sum $\text{mss}(A)$ is defined as:

$$\text{mss}(A) = \max_{1 \le i \le j \le m} \left(\sum_{k=i}^{j} x_k \right).$$

Now, given an integer array A of length n, Rikka wants you to calculate the sum of the maximum subsegment sums of all subsegments of A, i.e.

$$\sum_{1 \le i \le j \le n} \text{mss}([A_i, \ldots, A_j]).$$

Input

The first line contains a single integer n ($1 \le n \le 10^5$).

The second line contains n integers A_i ($-10^9 \le A_i \le 10^9$).

Output

Output a single line with a single integer, the answer. The answer can be very large, therefore, you are only required to output the answer modulo 2^{64}.

More formally, suppose the answer is x, you are required to find the smallest non-negative integer y satisfying $y = x + k \times 2^{64}$ for some integer k.

Examples

standard input	standard output
`5` `1 -1 1 -1 1`	`11`
`5` `1 -2 3 -4 5`	`39`
`10` `1 -3 -5 7 -9 10 8 -6 -4 2`	`555`
`4` `-1 -2 -3 -4`	`18446744073709551596`

Problem E. Rikka with Subsequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

As we all know, Rikka is not good at math. Yuta, her boyfriend, is worried about it. Therefore, he sets an interesting math problem For Rikka to practice.

Given a non-negative integer x, Rikka is required to find three non-negative integers a, b, c that satisfy the following three conditions:

1. $a + b = x$;
2. str(c) is a subsequence of str(a);
3. str(c) is a subsequence of str(b).

str(d) represents the decimal string representation of integer d. For example, str(0) ="0", str(103) ="103".

String $s = s_1 \dots s_n$ is a subsequence of string $t = t_1 \dots t_m$ if and only if there exists an index sequence $1 \le i_1 < i_2 < \cdots < i_n \le m$ satisfying $\forall j \in [1, n], s_j = t_{i_j}$.

To avoid the case of no solution, Yuta assumes there is a special choice "−" for c where str(−) is equal to the empty string. Under this assumption, $a = 0, b = 9, c = -$ becomes a valid solution of $x = 9$.

Finding a valid solution is an easy task even for Rikka. Therefore, Rikka wants to increase the difficulty: Rikka wants you to find a valid solution (a, b, c) so that the length of str(c) is as large as possible.

Input

The first line contains a single integer T ($1 \le T \le 10^4$), representing the number of test cases.

For each test case, the first line contains a single integer x ($|\text{str}(x)| \le 5000$).

The input guarantees that $\sum |\text{str}(x)| \le 10^5$.

Output

For each test case, output three lines, each with a single integer, representing a, b, c respectively.

If there are multiple optimal solutions, you need only to output any of them.

Example

standard input	standard output
3 2290283839620 1 9999999	1145141919810 1145141919810 1145141919810 0 1 - 4545454 5454545 454545

Problem G. Rikka with Game Theory

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Game theory is an interesting subject in computer science.

SG function is an important concept in game theory. Given a directed acyclic graph G_1 with vertex set V_1 and directed edge set E_1, for each vertex $u \in V_1$, its SG function $sg(u)$ is defined as:

$$sg(u) = \text{mex}(\{sg(v) | (u, v) \in E_1\})$$

where given a set S of non-negative integers, $\text{mex}(S)$ is defined as the smallest non-negative integer which is not in S.

Today, Rikka wants to generalize SG function to undirected graphs. Given an undirected graph G with vertex set V and undirected edge set E, a function f over V is a valid SG function on G if and only if:

- For each vertex $u \in V$, $f(u)$ is a non-negative integer;
- For each vertex $u \in V$, $f(u) = \text{mex}(\{f(v) | (u, v) \in E\})$.

Under this definition, there may be many valid SG functions for a graph. Therefore, Rikka wants to further figure out whether there is a connection between these valid SG functions. As the first step, your task is to calculate the number of valid SG functions for a given undirected graph G.

Input

The first line contains two integers n, m ($1 \le n \le 17, 0 \le m \le \frac{n(n-1)}{2}$), representing the number of vertices and edges in the graph.

Then m lines follow. Each line contains two integers u_i, v_i ($1 \le u_i, v_i \le n$), representing an edge in the graph.

The input guarantees that there are no self-loops and duplicate edges in the graph.

Output

Output a single line with a single integer, representing the number of valid SG functions.

Example

standard input	standard output
5 4 1 2 2 3 3 4 4 5	6

Note

For simplicity, we use list $[f(1), \dots, f(n)]$ to represent a function f.

For the sample input, there are 6 valid SG functions:

- $[0, 1, 0, 1, 0]$, $[0, 1, 2, 0, 1]$, $[0, 2, 1, 0, 1]$, $[1, 0, 1, 0, 1]$, $[1, 0, 1, 2, 0]$ and $[1, 0, 2, 1, 0]$.

Problem I. Rikka with RCPC

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Rikka is going to hold a worldwide competition, RCPC (Rikka's collegiate programming contest).

Preparing for a contest is time-consuming: It will take Rikka n days. Meanwhile, there are so many questions in the official QQ group of RCPC, raised by contestants, coaches, and even melon eating people. Therefore, Rikka decides to ignore the QQ group on some of the n days.

However, always ignoring questions may make contestants angry. Before the start of Day 1, the angry value A of contestants is 0. At the beginning of the i-th day, the angry value will be increased by a_i. Then:

- If A is strictly larger than a threshold T, contestants will be extremely angry, and Rikka will receive $2A$ points of attack. Then, at the end of this day, A will be cleared to 0;
- If A is no larger than T and Rikka chooses to ignore questions, nothing will happen on this day;
- If A is no larger than T and Rikka decides to answer questions, meanwhile, if Rikka ignores all questions on previous K days, i.e. from Day $i-K$ to Day $i-1$, contestants will feel the hardness of Rikka. Rikka will not receive any attack, and at the end of this day, A will be cleared to 0;
- Otherwise, even though Rikka chooses to answer questions, contestants will still blame Rikka for answering questions so slowly, and Rikka will receive A points of attack. At the end of this day, A will be cleared to 0.

Your task is to help Rikka to decide which days to answer questions so that the total attacks she received is minimized.

Input

The first line contains three integers n, K $(1 \le K < n \le 2 \times 10^5)$ and T $(1 \le T \le 10^9)$.

The second line contains n integers a_i $(1 \le a_i \le T)$.

Output

Output a single integer, the answer.

Examples

standard input	standard output
4 1 5 3 1 4 2	3
10 2 7 2 7 4 4 1 5 6 7 3 1	36

Note

For the first sample, one optimal plan is to answer questions on Day 1 and Day 3:

- On Day 1, the angry value is 3, and thus Rikka will receive 3 points of attacks.
- On Day 2, the angry value is 1, and thus nothing will happen;

- On Day 3, the angry value is 5. Since Rikka does not answer the questions on Day 2, contestants will feel the hardness of Rikka, and thus nothing will happen;
- On Day 4, the angry value is 2, and thus nothing will happen.

Problem J. Rikka with Book

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

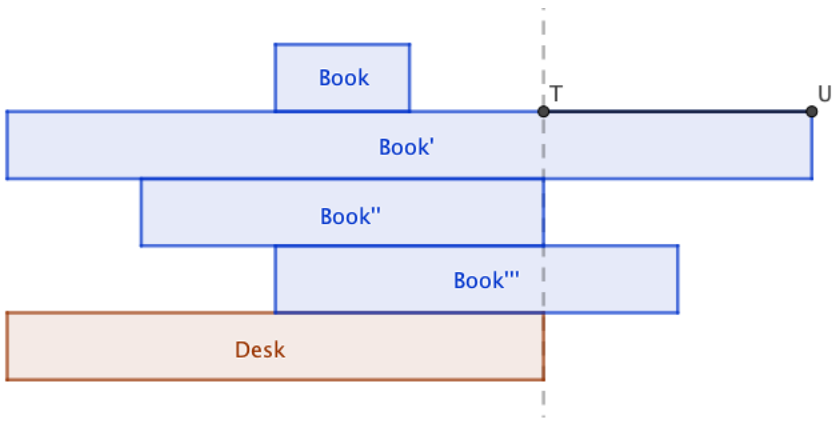
There are n books on Rikka's desk. The length, width, height and weight of the i-th book are l_i, 1, 1 and w_i respectively. We call the two sides of length l_i as long sides and two sides of length 1 as short sides. For each book, the mass is evenly distributed in its volume.

Today, Rikka wants to stack these books on her desk. Rikka plans to do it in n turns: In each turn, she will select out one book among all remaining books, and stack it directly above the previous book (The first book will be put directly on the desk).

For tidiness, Rikka makes the following four requirements:

- The left long sides of all books must be coplanar. So do the right long sides;
- The plane containing all left long sides must be vertical to the ground. So do the right long sides;
- All short sides of all books must be parallel to the border of the desk;
- The final book stack must be stable. Falling thick books are dangerous!

Informally, the books will be stacked as the following figure:



For fun, Rikka wants to make the book stack looking as strange as possible. For each book, Rikka defines its strangeness as the horizontal distance for its front short edge to be outside the desk. The strangeness of the whole book stack is defined as the maximum strangeness of books. For example, in the previous figure, the strangeness of the book stack is equal to the length of segment TU.

Rikka wants you to come up with a stacking plan so that the strangeness of the book stack is as large as possible.

More assumptions:

- Suppose the gravity coefficient is fixed. It will not change when the height change;
- Suppose the shape of the desk and books would never change.

Input

The first line contains a single integer n ($1 \le n \le 20$), the number of books.

The second line contains n integers l_i ($1 \le l_i \le 10^3$), representing the length of each book.

The third line contains n integers w_i ($1 \le w_i \le 10^3$), representing the weight of each book.

Output

Output a single line with a single real number, representing the maximum possible strangeness value.

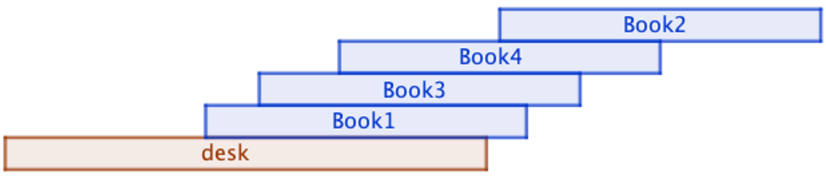
Your answer should have an absolute error or relative error of less than 10^{-9}.

Examples

standard input	standard output
4 2 2 2 2 1 1 1 1	2.08333333333
3 1 2 3 3 2 1	2.95833333333

Note

For the first sample, one optimal plan is shown in the following figure.



Here the strangeness of book $1, 2, 3, 4$ is $\frac{1}{4}, \frac{25}{12}, \frac{7}{12}, \frac{13}{12}$ respectively. Therefore, the strangeness of the book stack is $\frac{25}{12}$.

Problem K. Rikka with Composite Number

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Rikka is a professional problem setter. Today, she is going to generate test cases for a problem about Composite Number.

To randomly generate composite numbers, Rikka starts from a non-empty subset D of digits $\{1, 2, \ldots, 9\}$ and integer $c = 0$, and then generates in turns. In each turn:

1. Rikka selects a digit d from D uniformly at random, and then changes c to $c \times 10 + d$;
2. If c has already been a composite integer, Rikka takes c as the result. Otherwise, Rikka returns to Step 1 and starts a new turn.

The time cost of a generator is crucial. Therefore, Rikka wants you to calculate the expected number of the turns used by the generator to generate a composite number.

A positive integer n is a composite integer if and only if there exists an integer $k \in [2, n-1]$ satisfying k is a factor of n.

Input

The first line contains a 01-string of length 9. The i-th character is 1 if and only if digit i is inside D.

The input guarantees that D is not empty.

Output

Output a single integer, representing the expected number of turns.

The answer is guaranteed to be a rational number. You are required to output the answer module 998244353. Formally, if the simplest fraction representation of the answer is $\frac{x}{y}$, you need to output $x \times y^{998244351} \bmod 998244353$.

Examples

standard input	standard output
100000000	3
001100000	499122178

Note

For the first sample, the generator must return 111 in the third turn.

For the second sample, there are 3 possibilities:

- Return 4 in the first turn, with probability $\frac{1}{2}$;
- Return 33 in the second turn, with probability $\frac{1}{4}$;
- Return 34 in the second turn, with probability $\frac{1}{4}$.

Therefore, the expected number of turns is $\frac{3}{2}$.

Problem N. NFC Message

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are trying to send a message via NFC to a friend from your smartphone. A smartphone can connect to another one nearby via NFC if the distance between them is not greater than D.

Determine whether a communication path between phones can be formed via zero or more intermediate smartphones so that your message can be successfully transmitted or not.

Input

The first line consists of two positive integers N ($1 \le N \le 10$) and D ($1 \le D \le 10$); N is the number of smartphones around, and D the transmission threshold. The next N lines are the X and Y coordinates of the locations of the N smartphones, of which the first one is your phone's, and the last one your friend's. All X and Y are non-negative integers does not exceeding 100.

Output

The (lower-case) character '`y`' if there is a connected path from your phone to your friend's, or '`n`' if there is not.

Example

standard input	standard output
4 7 1 4 6 2 9 7 14 4	y
5 6 7 1 5 5 1 6 8 7 20 15	n

Problem O. Ordered Sequences

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider sequences of 'A' and 'B', starting with 'B'. Lets sort them in next order: shortest sequences are coming first, if two sequences have the same length, lexicographically smallest is coming first and then lets number them starting from 1.

Given the sequence, find its number.

Input

First line of the input contains one integer n ($1 \leq n \leq 1000$) — number of test cases. Then n lines follow, each contains non-empty string containing only of 'A' and 'B', starting from 'B' and no more than 24 characters long — the given sequence.

Output

For each test case print number of the given sequence.

Example

standard input	standard output
3 B BA BB	1 2 3

Problem P. Performing Shuffle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Define *shuffle* of a list of words as dividing the list exactly in half, and then alternating words from the two halves, starting with the top half.

Given a list of words, perform a shuffle. If there is an odd number of words, give the top half split one more card than the bottom half.

Input

There will be several test cases in the input. Each test case will begin with a line with a single integer n ($1 \le n \le 1,000$), indicating the number of words. On each of the next n lines will be a string from 1 to 80 characters in length, which is the word. It will contain only capital letters and dashes.

Within a test case, all words will be unique. Input will end with a line with a single 0.

Output

For each test case, output n lines, consisting of the list after a shuffle.

Examples

standard input	standard output
4 LAZY BROWN FOX DOG 5 ONE TWO THREE FOUR FIVE 0	LAZY FOX BROWN DOG FOX ONE FOUR TWO FIVE THREE

Problem Q. Quick Enough?

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given a distance (in km) and two speeds (in km per hour), determine the difference in time to travel the distance.

Input

There will be several test cases in the input. Each test case will consist of three integers on a single line, d ($1 \le d \le 10\,000$), s_1 and s_2 ($1 \le s1 < s2 \le 1000$), where d is a distance, and s_1 and s_2 are speeds. The integers will be separated by a single space, and there will be no leading or trailing blanks. The input will end with a line with three 0s.

Output

For each test case, output a time in the form $H : MM : SS$. Minutes (MM) and Seconds (SS) should be exactly two characters long, padded with a 0 if necessary. Seconds should be rounded. Hours should use the minimum number of digits necessary.

Examples

standard input	standard output
21 75 80 26 90 130 0 0 0	0:01:03 0:05:20

Problem R. Rolling On Text

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Take some text. Put a small ball at the top of the first letter of the first word of the first sentence. The ball is drawn via gravity downwards. The text is also at a slight angle, so the ball wants to also move towards the right. The ball can freely move between the lines, and can drop through spaces. Considering the first column to be column 1, what column will the ball end up in?

Input

There will be several test cases in the input. Each test case will begin with an integer n ($1 \le n \le 1,000$) on its own line, indicating the number of lines of text. On each of the next n lines will be text, consisting of printable ASCII characters and spaces. There will be no tabs, nor any other unprintable characters. Each line will be between 1 and 100 characters long. The input will end with a line containing a single 0.

Output

For each test case, output a single integer on its own line, indicating the column from which the ball will drop. Output no spaces, and do not separate answers with blank lines.

Example

standard input

```
7
Take some text. Put a small ball at the top of the first letter of
the first word of the first sentence.  The ball  is drawn  via  gravity
downwards.
The  text  is  also  at a slight  angle,  so the  ball  wants  to  also
move  towards  the  right.  The  ball  can  freely  move  between  the lines,
and  can  drop  through  spaces. Considering  the  first  column  to
be  column  1,  what column will the ball end up in?
3
Thisisverylongword
shorter
more
0
```

standard output

```
15
19
```

Problem S. Sum Them All

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Given three integers A, K, P, you are asked to calculate the value of the following sum.

$$\sum_{i=1}^{K} A^i mod P$$

Input

The first and only line of the input contains 3 integers, A ($0 \le A \le 10^8$), K ($1 \le K \le 10^{16}$) and P ($1 \le P \le 10^8$), separated by a space.

Output

The only line in the output contains a single integer, the value of the summation.

Examples

standard input	standard output
3 4 101	19