TASK	BELA	PUTOVANJE	PIANINO	PAROVI	KRUMPIRKO	SAN
input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 seconds	1 second	1 second	1 second	1 second	5 seconds
memory limit	64 MB	64 MB	64 MB	64 MB	64 MB	64 MB
score	**50**	**80**	**100**	**120**	**140**	**160**
	total 650					

Young Mirko is a smart, but mischievous boy who often wanders around parks looking for new ideas. This time he's come across pensioners playing the card game Belote. They've invited him to help them determine the total number of points won in a single game.

Each card can be uniquely determined by its symbol and suit. A set of four cards is called a **hand**. In each game, one suit that "trumps" any other and is called the **dominant suit**. The number of points in a single game is equal to the sum of **scoring values** of each card from each hand won in the game. Mirko has noticed that the pensioners have won N hands and that suit B was the dominant suit.

The scoring values of cards are given in the following table:

Card symbol	**Card scoring value**	
	If dominant suit	**If not dominant suit**
A	11	11
K	4	4
Q	3	3
J	20	2
T	10	10
9	14	0
8	0	0
7	0	0

Write a programme that will determine and output the total number of points won in the game.

INPUT

The first line contains the number of hands N ($1 \leqslant N \leqslant 100$) and the value of suit B (S, H, D, C) from the task.
Each of the following $4N$ lines contains the description of card K_i (the first character being the label of the i^{th} card $(A, K, Q, J, T, 9, 8, 7)$, the second character being the suit of the i^{th} card (S, H, D, C)).

OUTPUT

The first and only line of output must contain the number of points from the task.

SAMPLE TESTS

input

```
2 S
TH
9C
KS
QS
JS
TD
AD
JH
```

output

```
60
```

input

```
4 H
AH
KH
QH
JH
TH
9H
8H
7H
AS
KS
QS
JS
TS
9S
8S
7S
```

output

```
92
```

Clarification of the second example: The total number of points is equal to $11 + 4 + 3 + 20 + 10 + 14 + 0 + 0 + 11 + 4 + 3 + 2 + 10 + 0 + 0 + 0 = 92$ points.

Young Mislav loves spending time in nature and, most of all, he loves spending time in forests. The fresh air and lovely sounds make the forest his favourite location. Mislav has decided to spend this afternoon in a forest and, because he's so practical, he's also decided to stuff himself with food. His belly can contain C amount of food.

He will have the opportunity to eat various fruits of nature (mushrooms, chestnuts, berries, and so on) while walking through the forest. All fruits are mutually different given their type and he'd like to eat as much different fruits as possible, but with the condition that he doesn't overeat. In other words, the total weight of the fruits he's eaten must not be larger than C. Also, when Mislav decides to start eating, he tries to eat every next fruit **if it's possible to eat it and not overeat**. In the case when he doesn't have the capacity to eat it, he just moves on.

An array of weights of N fruits represents the weight and order of fruits that Mislav came across in the forest. Determine the maximum amount of different fruits that Mislav can eat.

INPUT

The first line of input contains two integers N and C ($1 \leqslant N \leqslant 1\,000, 1 \leqslant C \leqslant 1\,000\,000$) from the task.
The second line contains N integers w_i ($1 \leqslant w_i \leqslant 1000$) that represent the fruits' weight.

OUTPUT

The first and only line of output must contain the maximum possible amount of different fruits that Mislav can eat.

SAMPLE TESTS

input	**input**	**input**
5 5 3 1 2 1 1	7 5 1 5 4 3 2 1 1	5 10 3 2 5 4 3
output	**output**	**output**
4	3	3

Clarification of the first example: If Mislav decides to start eating from fruit (3), then he will have eaten 3 different fruits (3, 1, 1). If he starts eating from fruit (1), he will have eaten 4 fruits (1, 2, 1, 1).

Young Mirka is an amateur musician. She plays the multi-piano. A multi-piano consists of an infinite number of multi-keys, denoted with integers that can be interpreted as the pitch. A multi-composition (a composition written for a multi-piano) can be represented with a finite array of integers, where integers denote the order of multi-keys to press in order to play the multi-composition.

Young Mirka has heard a multi-composition on the multi-radio and now she wants to play it. Unfortunately, she cannot hear exactly which key was pressed, but instead she can hear whether the pressed multi-key was higher, lower or equal to the previously played key (a higher key is denoted with a larger number). Therefore she has decided to play the composition in the following way:

- before playing, she will choose one non-negative integer K
- in the beginning, she will play the correct multi-key (her multi-teacher told her which multi-key that is)
- when she hears that the multi-key played in the multi-composition is higher than the previous multi-key played in the multi-composition, she will play the multi-key denoted with the integer larger than the multi-key she played previously by K
- analogously, when she hears that the multi-key played in the multi-composition is lower than the previous multi-key played in the multi-composition, she will play the multi-key denoted with the integer smaller than the multi-key she played previously by K
- when she hears that the multi-key played in the multi-composition is equal to the previous multi-key played in the multi-composition, she will repeat the multi-key she played previously

Notice that, when Mirka is playing, she does not compare the pitch of the keys she played to the pitch of the keys from the composition.

Help Mirka choose the integer K in order to hit as many correct pitches as possible.

INPUT

The first line of input contains the integer N ($2 \leqslant N \leqslant 10^6$), the number of multi-keys in the multi-composition on the multi-radio.
The second line of input contains N integers a_i ($-10^9 \leqslant a_i \leqslant 10^9$), the multi-keys played in the multi-composition.

OUTPUT

The first line of output must contain the maximum number of multi-keys that Mirka can play correctly.
The second line of output must contain the non-negative number K that Mirka must choose in order to hit as many correct pitches as possible. The number must be smaller than or equal to $2 \cdot 10^9$.

Please note: The required number does not have to be unique, but will surely exist within the given constraints.

SAMPLE TESTS

input	**input**
5 1 2 0 3 1	7 2 1 -6 -2 1 6 10
output	**output**
3 2	5 4

Clarification of the first example: Mirka will play the following keys, respectively: **1**, 2, 0, **3**, **1**. Denoted in bold are the keys that she played correctly.
Clarification of the second example: Mirka will play the following keys, respectively: **2**, -2, **-6**, **-2**, 2, **6**, **10**.

Mirko and Slavko are playing a game. Mirko's turn is first and he chooses a **non-empty** set of pairs of numbers between 1 and N (inclusive) under the condition that the numbers that comprise a pair are mutually **relatively prime**. The numbers that comprise a pair must be different. For example, for $N = 5$, Mirko could have chosen the following set of pairs: $\{\{1, 2\}, \{3, 4\}, \{2, 5\}, \{3, 5\}\}$.

Slavko's turn is second and his goal is to find a **partition** for Mirko's set of pairs. Mirko's set of pairs has a **partition** if an integer x from the set $\{2, 3, ..., N\}$ exists such that, for each pair $\{a, b\}$, one of the following holds:

- $a, b < x$
- $a, b \geq x$

For example, a set of pairs $\{\{1, 2\}, \{3, 4\}\}$ has a partition $x = 3$. If a partition exists, Slavko will surely find it.

Mirko wins if Slavko can't find a partition for his set. Determine how many different sets of pairs exists that Mirko can initially choose and be sure of his victory. Given the fact that the total number of sets can be very large, output the number modulo 1 000 000 000.

INPUT

The first line of input contains the integer N ($1 \leqslant N \leqslant 20$).

OUTPUT

The first and only line of output must contain the required number.

SAMPLE TESTS

input	**input**	**input**
2	3	4
output	**output**	**output**
1	5	21

Clarification of the first example: The only set of pairs that meets the given requirements is $\{\{1, 2\}\}$.
Clarification of the second example: An example of a set that meets the given requirements is $\{\{1, 3\}, \{1, 2\}\}$.

Young Mr. Potato is opening two new stores where he will, you guessed it, sell potatoes. Mr. Potato gets his potatoes from N farmers. Each farmer offers **exactly** a_i potatoes per bag for a total price of c_i. Mr. Potato is going to buy all bags of potatoes from all farmers and place the bags in his two stores.

Let's denote the average potato price in the first store with P_1, and the average potato price in the second store with P_2. The average potato price in a store is equal to the **ratio of the price and the total number of potatoes** in the store. Taking into account logistical difficulties and the amount of potatoes in the stores, he wants the product of the average prices of potatoes in the stores to be minimal. In other words, he wants the product of P_1 and P_2 to be minimal.

After Mr. Potato settles on a division of bags in the stores, at least one store must have exactly L **bags**.

INPUT

The first line of input contains two integers N and L ($2 \leqslant N \leqslant 100$, $1 \leqslant L < N$), the number of potato bags and the number of potato bags in at least one store.
The second line of input contains N integers a_i ($1 \leqslant a_i \leqslant 100$), separated by space.
The third line of input contains N integers c_i ($1 \leqslant c_i \leqslant 1\,000\,000$), separated by space.
The sum of all a_i will be $\leqslant 500$.

OUTPUT

The first and only line of output must contain the minimal product of P_1 and P_2 from the task, rounded to three decimal places.

SCORING

In at least 30% of examples, it will hold $N \leqslant 20$.

SAMPLE TESTS

input

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

output

```
0.556
```

input

```
3 2
2 2 2
3 3 3
```

output

```
2.250
```

Anica is having peculiar dream. She is dreaming about an infinite board. On that board, an infite table consisting of infinite rows and infinite columns containing infinite numbers is drawn. Interestingly, each number in the table appears a finite number of times.

The table is of exceptionally regular shape and its values meet the requirements of a simple recursive relation. The first cell of each row contains the ordinal number of that row. A value of a cell that is not in the first column can be calculated by adding up the number in the cell to the left of it and that same number, only written in reverse (in decimal representation).

Formally, if $A(i, j)$ denotes the value in the i^{th} row and the j^{th} column, it holds:

- $A(i, 1) = i$
- $A(i, j) = A(i, j-1) + \text{rev}^1(A(i, j-1))$, for each $j > 1$

1	2	4	8	16	77	154	
2	4	8	16	77	154	605	
3	6	12	33	66	132	363	...
4	8	16	77	154	605	1111	
5	10	11	22	44	88	176	
⋮							⋱

The first few rows and columns of the table.
Notice that the table is infinite only in 2 directions.

Anica hasn't shown too much interest in the board and obliviously passed by it. Behind the board, she noticed a lamp that immediately caught her attention. Anica also caught the lamp's attention, so the friendly ghost Božo came out of it.

"Anica! If you answer correctly to my Q queries, you will win a package of Dorina wafer or Domaćica cookies, based on your own choice! I wouldn't want to impose my stance, but in my personal opinion, the Dorina wafer cookies are better. Each query will consist of two integers A and B. You must answer how many appearances of numbers from the interval $[A, B]$ there are on the board."

Unfortunately, Anica couldn't give an answer to the queries and woke up.

"Ah, I didn't win the Dorina cookies, but at least I have a task for COCI", she thought and went along with her business.

INPUT

The first line of input contains the integer Q ($1 \leqslant Q \leqslant 10^5$), the number of queries.
Each of the following Q lines contains two integers A and B ($1 \leqslant A \leqslant B \leqslant 10^{10}$) that represent the interval from the query.

OUTPUT

The i^{th} line of output must contain a single integer – the answer to the i^{th} query.

SCORING

In test cases worth 50% of total points, it will hold ($1 \leqslant A, B \leqslant 10^6$).

[1] rev(x) denotes the number x written in reverse in decimal representation. For example, rev(213) = 312, rev(406800) = 008604 = 8604.

SAMPLE TESTS

input	**input**	**input**
2 1 10 5 8	3 17 144 121 121 89 98	1 1 1000000000
output	**output**	**output**
18 8	265 25 10	1863025563