A - Takahashi san 2

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

KEYENCE has a culture of addressing everyone with the suffix "-san," regardless of roles, age, or positions.

You are given a string S consisting of lowercase English letters.
If S ends with `san`, print `Yes`; otherwise, print `No`.

Constraints

- S is a string of length between 4 and 30, inclusive, consisting of lowercase English letters.

Input

The input is given from Standard Input in the following format:

```
S
```

Output

If S ends with `san`, print `Yes`; otherwise, print `No`.

Sample Input 1

```
takahashisan
```

Sample Output 1

```
Yes
```

The string $S =$ `takahashisan` ends with `san`, so print `Yes`.

Sample Input 2

```
aokikun
```

Sample Output 2

```
No
```

The string $S =$ `aokikun` does not end with `san`, so print `No`.

B - Unvarnished Report

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

> KEYENCE has a culture of reporting things as they are, whether good or bad.
> So we want to check whether the reported content is exactly the same as the original text.

You are given two strings S and T, consisting of lowercase English letters.
If S and T are equal, print 0; otherwise, print the position of the first character where they differ.
Here, if the i-th character exists in only one of S and T, consider that the i-th characters are different.

More precisely, if S and T are not equal, print the smallest integer i satisfying one of the following conditions:

- $1 \leq i \leq |S|, 1 \leq i \leq |T|$, and $S_i \neq T_i$.
- $|S| < i \leq |T|$.
- $|T| < i \leq |S|$.

Here, $|S|$ and $|T|$ denote the lengths of S and T, respectively, and S_i and T_i denote the i-th characters of S and T, respectively.

Constraints

- S and T are strings of length between 1 and 100, inclusive, consisting of lowercase English letters.

Input

The input is given from Standard Input in the following format:

S
T

Output

If S and T are equal, print 0; otherwise, print the position of the first character where they differ.

Sample Input 1

```
abcde
abedc
```

Sample Output 1

```
3
```

We have $S =$ `abcde` and $T =$ `abedc`.

S and T have the same first and second characters, but differ at the third character, so print 3.

Sample Input 2

```
abcde
abcdefg
```

Sample Output 2

```
6
```

We have $S =$ `abcde` and $T =$ `abcdefg`.

S and T are equal up to the fifth character, but only T has a sixth character, so print 6.

Sample Input 3

```
keyence
keyence
```

Sample Output 3

```
0
```

S and T are equal, so print 0.

C - Separated Lunch

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

As KEYENCE headquarters have more and more workers, they decided to divide the departments in the headquarters into two groups and stagger their lunch breaks.

KEYENCE headquarters have N departments, and the number of people in the i-th department $(1 \leq i \leq N)$ is K_i.

When assigning each department to Group A or Group B, having each group take lunch breaks at the same time, and ensuring that the lunch break times of Group A and Group B do not overlap, find the minimum possible value of the maximum number of people taking a lunch break at the same time.
In other words, find the minimum possible value of the larger of the following: the total number of people in departments assigned to Group A, and the total number of people in departments assigned to Group B.

Constraints

- $2 \leq N \leq 20$
- $1 \leq K_i \leq 10^8$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
K_1 \quad K_2 \quad \ldots \quad K_N
\end{array}
$$

Output

Print the minimum possible value of the maximum number of people taking a lunch break at the same time.

Sample Input 1

```
5
2 3 5 10 12
```

Sample Output 1

```
17
```

When assigning departments $1, 2$, and 5 to Group A, and departments 3 and 4 to Group B, Group A has a total of $2 + 3 + 12 = 17$ people, and Group B has a total of $5 + 10 = 15$ people. Thus, the maximum number of people taking a lunch break at the same time is 17.

It is impossible to assign the departments so that both groups have 16 or fewer people, so print 17.

Sample Input 2

```
2
1 1
```

Sample Output 2

```
1
```

Multiple departments may have the same number of people.

Sample Input 3

```
6
22 25 26 45 22 31
```

Sample Output 3

```
89
```

For example, when assigning departments $1, 4$, and 5 to Group A, and departments $2, 3$, and 6 to Group B, the maximum number of people taking a lunch break at the same time is 89.

D - Laser Marking

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 350 points

Problem Statement

There is a printing machine that prints line segments on the xy-plane by emitting a laser.

- At the start of printing, the laser position is at coordinate $(0, 0)$.
- When printing a line segment, the procedure below is followed.
  - First, move the laser position to one of the endpoints of the line segment.
    - One may start drawing from either endpoint.
  - Then, move the laser position in a straight line from the current endpoint to the other endpoint while emitting the laser.
    - It is not allowed to stop printing in the middle of a line segment.
- When not emitting the laser, the laser position can move in any direction at a speed of S units per second.
- When emitting the laser, the laser position can move along the line segment being printed at a speed of T units per second.
- The time required for operations other than moving the laser position can be ignored.

Takahashi wants to print N line segments using this printing machine.
The i-th line segment connects coordinates (A_i, B_i) and (C_i, D_i).
Some line segments may overlap, in which case he needs to print the overlapping parts for each line segment separately.

What is the minimum number of seconds required to complete printing all the line segments when he operates the printing machine optimally?

Constraints

- All input values are integers.
- $1 \leq N \leq 6$
- $1 \leq T \leq S \leq 1000$
- $-1000 \leq A_i, B_i, C_i, D_i \leq 1000$
- $(A_i, B_i) \neq (C_i, D_i)$ ($1 \leq i \leq N$)

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad S \quad T \\
A_1 \quad B_1 \quad C_1 \quad D_1 \\
\vdots \\
A_N \quad B_N \quad C_N \quad D_N
\end{array}
$$

Output

Print the answer.

Your output will be considered correct if the absolute or relative error from the true value does not exceed 10^{-6}.

Sample Input 1

```
3 2 1
1 3 2 1
0 2 0 0
3 0 2 0
```

Sample Output 1

```
6.44317475868633722080
```

- Emit the laser while moving the laser position from $(0,0)$ to $(0,2)$, printing the second line segment.
  - This takes 2 seconds.
- Move the laser position from $(0,2)$ to $(1,3)$ without emitting the laser.
  - This takes $\sqrt{2}/2$ seconds.
- Emit the laser while moving the laser position from $(1,3)$ to $(2,1)$, printing the first line segment.
  - This takes $\sqrt{5}$ seconds.
- Move the laser position from $(2,1)$ to $(2,0)$ without emitting the laser.
  - This takes $1/2$ second.
- Emit the laser while moving the laser position from $(2,0)$ to $(3,0)$, printing the third line segment.
  - This takes 1 second.
- The total time taken is $2+(\sqrt{2}/2)+\sqrt{5}+(1/2)+1 \approx 6.443175$ seconds.

Sample Input 2

```
2 1 1
0 0 10 10
0 2 2 0
```

Sample Output 2

```
20.97056274847714058517
```

Sample Input 3

```
6 3 2
-1000 -1000 1000 1000
1000 -1000 -1000 1000
-1000 -1000 1000 1000
1000 -1000 -1000 1000
1000 1000 -1000 -1000
-1000 1000 1000 -1000
```

Sample Output 3

```
9623.35256169626864153344
```

Multiple line segments overlap here, and you need to print the overlapping parts for each line segment separately.

Sample Input 4

```
6 10 8
1000 1000 -1000 -1000
1000 -1000 -1000 -1000
-1000 1000 1000 1000
-1000 1000 -1000 -1000
1000 1000 1000 -1000
1000 -1000 -1000 1000
```

Sample Output 4

```
2048.52813742385702910909
```

E - Sensor Optimization Dilemma 2

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 475 points

Problem Statement

The manufacturing of a certain product requires N processes numbered $1, 2, \dots, N$.

For each process i, there are two types of machines S_i and T_i available for purchase to handle it.

- Machine S_i: Can process A_i products per day per unit, and costs P_i yen per unit.
- Machine T_i: Can process B_i products per day per unit, and costs Q_i yen per unit.

You can purchase any number of each machine, possibly zero.

Suppose that process i can handle W_i products per day as a result of introducing machines.

Here, we define the production capacity as the minimum of W, that is, $\displaystyle \min_{i=1}^{N} W_i$.

Given a total budget of X yen, find the maximum achievable production capacity.

Constraints

- All input values are integers.
- $1 \le N \le 100$
- $1 \le A_i, B_i \le 100$
- $1 \le P_i, Q_i, X \le 10^7$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & X & & \\
A_1 & P_1 & B_1 & Q_1 \\
A_2 & P_2 & B_2 & Q_2 \\
\vdots & & & \\
A_N & P_N & B_N & Q_N
\end{array}
$$

Output

Print the answer as an integer.

Sample Input 1

```
3 22
2 5 3 6
1 1 3 3
1 3 2 4
```

Sample Output 1

```
4
```

For example, by introducing machines as follows, we can achieve a production capacity of 4, which is the maximum possible.

- For process 1, introduce 2 units of machine S_1.
  - This allows processing 4 products per day and costs a total of 10 yen.
- For process 2, introduce 1 unit of machine S_2.
  - This allows processing 1 product per day and costs a total of 1 yen.
- For process 2, introduce 1 unit of machine T_2.
  - This allows processing 3 products per day and costs a total of 3 yen.
- For process 3, introduce 2 units of machine T_3.
  - This allows processing 4 products per day and costs a total of 8 yen.

Sample Input 2

```
1 10000000
100 1 100 1
```

Sample Output 2

```
1000000000
```

Sample Input 3

```
1 1
1 10000000 1 10000000
```

Sample Output 3

```
0
```

There may be cases where a positive production capacity cannot be achieved.

Sample Input 4

```
10 7654321
8 6 9 1
5 6 4 3
2 4 7 9
7 8 9 1
7 9 1 6
4 8 9 1
2 2 8 9
1 6 2 6
4 2 3 4
6 6 5 2
```

Sample Output 4

```
894742
```

F - Shipping

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 550 points

Problem Statement

> KEYENCE is famous for quick delivery.

In this problem, the calendar proceeds as Day 1, Day 2, Day 3, $\dots$.

There are orders $1, 2, \dots, N$, and it is known that order i will be placed on Day T_i.
For these orders, shipping is carried out according to the following rules.

- At most K orders can be shipped together.
- Order i can only be shipped on Day T_i or later.
- Once a shipment is made, the next shipment cannot be made until X days later.
  - That is, if a shipment is made on Day a, the next shipment can be made on Day $a + X$.

For each day that passes from order placement to shipping, dissatisfaction accumulates by 1 per day.
That is, if order i is shipped on Day S_i, the dissatisfaction accumulated for that order is $(S_i - T_i)$.

Find the minimum possible total dissatisfaction accumulated over all orders when you optimally schedule the shipping dates.

Constraints

- All input values are integers.
- $1 \leq K \leq N \leq 100$
- $1 \leq X \leq 10^9$
- $1 \leq T_1 \leq T_2 \leq \dots \leq T_N \leq 10^{12}$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & K & X & \\
T_1 & T_2 & \ldots & T_N
\end{array}
$$

Output

Print the answer as an integer.

Sample Input 1

```
5 2 3
1 5 6 10 12
```

Sample Output 1

```
2
```

For example, by scheduling shipments as follows, we can achieve a total dissatisfaction of 2, which is the minimum possible.

- Ship order 1 on Day 1.
  - This results in dissatisfaction of $(1-1)=0$, and the next shipment can be made on Day 4.
- Ship orders 2 and 3 on Day 6.
  - This results in dissatisfaction of $(6-5)+(6-6)=1$, and the next shipment can be made on Day 9.
- Ship order 4 on Day 10.
  - This results in dissatisfaction of $(10-10)=0$, and the next shipment can be made on Day 13.
- Ship order 5 on Day 13.
  - This results in dissatisfaction of $(13-12)=1$, and the next shipment can be made on Day 16.

Sample Input 2

```
1 1 1000000000
1000000000000
```

Sample Output 2

```
0
```

Sample Input 3

```
15 4 5
1 3 3 6 6 6 10 10 10 10 15 15 15 15 15
```

Sample Output 3

```
35
```

G - Only One Product Name

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 600 points

Problem Statement

All KEYENCE product names consist of **two uppercase English letters**.
They have already used N product names, the i-th of which $(1 \leq i \leq N)$ is S_i.
Once a product name is used, it cannot be reused, so they decided to create an NG (Not Good) list to quickly identify previously used product names.

The NG list must satisfy the following conditions.

- It consists of one or more strings, each consisting of uppercase English letters.
- For each already used product name, there exists at least one string in the list that contains the name as a (contiguous) substring.
- None of the strings in the list contain any length-2 (contiguous) substring that is not an already used product name.

Find the minimum possible number of strings in the NG list.

Constraints

- $1 \leq N \leq 26^2$
- N is an integer.
- Each S_i is a string of length 2 consisting of uppercase English letters.
- All $S_1, S_2, \ldots, S_N$ are distinct.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
S_1 \\
S_2 \\
\vdots \\
S_N
\end{array}
$$

Output

Print the minimum possible number of strings in the NG list.

Sample Input 1

```
7
AB
BC
CA
CD
DE
DF
XX
```

Sample Output 1

```
3
```

One NG list satisfying the conditions is the one consisting of the following three strings:

- CABCDE
- DF
- XX

This has three strings, and there is no NG list satisfying the conditions with 2 or fewer strings, so print 3.

Sample Input 2

```
5
AC
BC
CD
DE
DF
```

Sample Output 2

```
2
```

One NG list satisfying the conditions is the one consisting of the following two strings:

- `ACDE`
- `BCDF`

Note that each used product name may appear in multiple strings in the NG list or multiple times within the same string.

Sample Input 3

```
6
AB
AC
CB
AD
DB
BA
```

Sample Output 3

```
1
```

For example, an NG list consisting only of `ABACBADB` satisfies the conditions.