A. Lucky Year

1 second, 256 megabytes

Apart from having lots of holidays throughout the year, residents of Berland also have whole *lucky* years. Year is considered *lucky* if it has no more than 1 non-zero digit in its number. So years 100, 40000, 5 are *lucky* and 12, 3001 and 12345 are not.

You are given current year in Berland. Your task is to find how long will residents of Berland wait till the next *lucky* year.

Input

The first line contains integer number n ($1 \le n \le 10^9$) — current year in Berland.

Output

Output amount of years from the current year to the next *lucky* one.

input
```
4
```
output
```
1
```

input
```
201
```
output
```
99
```

input
```
4000
```
output
```
1000
```

In the first example next *lucky* year is 5. In the second one — 300. In the third — 5000.

B. Average Sleep Time

1 second, 256 megabytes

It's been almost a week since Polycarp couldn't get rid of insomnia. And as you may already know, one week in Berland lasts k days!

When Polycarp went to a doctor with his problem, the doctor asked him about his sleeping schedule (more specifically, the average amount of hours of sleep per week). Luckily, Polycarp kept records of sleep times for the last n days. So now he has a sequence $a_1, a_2, ..., a_n$, where a_i is the sleep time on the i-th day.

The number of records is so large that Polycarp is unable to calculate the average value by himself. Thus he is asking you to help him with the calculations. To get the average Polycarp is going to consider k consecutive days as a week. So there will be $n - k + 1$ weeks to take into consideration. For example, if $k = 2$, $n = 3$ and $a = [3, 4, 7]$, then the result is $\frac{(3+4)+(4+7)}{2} = 9$.

You should write a program which will calculate average sleep times of Polycarp over all weeks.

Input

The first line contains two integer numbers n and k ($1 \le k \le n \le 2 \cdot 10^5$).

The second line contains n integer numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^5$).

Output

Output average sleeping time over all weeks.

The answer is considered to be correct if its absolute or relative error does not exceed 10^{-6}. In particular, it is enough to output real number with at least 6 digits after the decimal point.

input
```
3 2
3 4 7
```

output

```
9.0000000000
```

input

```
1 1
10
```

output

```
10.0000000000
```

input

```
8 2
1 2 4 100000 123 456 789 1
```

output

```
28964.2857142857
```

In the third example there are $n - k + 1 = 7$ weeks, so the answer is sums of all weeks divided by 7.

C. Tea Party

1 second, 256 megabytes

Polycarp invited all his friends to the tea party to celebrate the holiday. He has n cups, one for each of his n friends, with volumes $a_1, a_2, ..., a_n$. His teapot stores w milliliters of tea ($w \le a_1 + a_2 + ... + a_n$). Polycarp wants to pour tea in cups in such a way that:

- Every cup will contain tea for at least half of its volume
- Every cup will contain integer number of milliliters of tea
- All the tea from the teapot will be poured into cups
- All friends will be *satisfied*.

Friend with cup i won't be *satisfied*, if there exists such cup j that cup i contains less tea than cup j but $a_i > a_j$.

For each cup output how many milliliters of tea should be poured in it. If it's impossible to pour all the tea and satisfy all conditions then output -1.

Input

The first line contains two integer numbers n and w ($1 \le n \le 100$, $1 \le w \le \sum_{i=1}^{n} a_i$).

The second line contains n numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 100$).

Output

Output how many milliliters of tea every cup should contain. If there are multiple answers, print any of them.

If it's impossible to pour all the tea and satisfy all conditions then output -1.

input

```
2 10
8 7
```

output

```
6 4
```

input

```
4 4
1 1 1 1
```

output

```
1 1 1 1
```

input

```
3 10
9 8 10
```

output

```
-1
```

In the third example you should pour to the first cup at least 5 milliliters, to the second one at least 4, to the third one at least 5. It sums up to 14, which is greater than 10 milliliters available.

D. Array Division

2 seconds, 256 megabytes

Vasya has an array a consisting of positive integer numbers. Vasya wants to divide this array into two non-empty consecutive parts (the prefix and the suffix) so that the sum of all elements in the first part equals to the sum of elements in the second part. It is not always possible, so Vasya will move some element before dividing the array (Vasya will erase some element and insert it into an arbitrary position).

Inserting an element in the same position he was erased from is also considered moving.

Can Vasya divide the array after choosing the right element to move and its new position?

Input

The first line contains single integer n $(1 \le n \le 100000)$ — the size of the array.

The second line contains n integers $a_1, a_2 ... a_n$ $(1 \le a_i \le 10^9)$ — the elements of the array.

Output

Print `YES` if Vasya can divide the array after moving one element. Otherwise print `NO`.

input
```
3
1 3 2
```
output
```
YES
```

input
```
5
1 2 3 4 5
```
output
```
NO
```

input
```
5
2 2 3 4 5
```
output
```
YES
```

In the first example Vasya can move the second element to the end of the array.

In the second example no move can make the division possible.

In the third example Vasya can move the fourth element by one position to the left.

E. Selling Souvenirs

2 seconds, 256 megabytes

After several latest reforms many tourists are planning to visit Berland, and Berland people understood that it's an opportunity to earn money and changed their jobs to attract tourists. Petya, for example, left the IT corporation he had been working for and started to sell souvenirs at the market.

This morning, as usual, Petya will come to the market. Petya has n different souvenirs to sell; ith souvenir is characterised by its weight w_i and cost c_i. Petya knows that he might not be able to carry all the souvenirs to the market. So Petya wants to choose a subset of souvenirs such that its total weight is not greater than m, and total cost is maximum possible.

Help Petya to determine maximum possible total cost.

Input

The first line contains two integers n and m $(1 \le n \le 100000,$ $1 \le m \le 300000)$ — the number of Petya's souvenirs and total weight that he can carry to the market.

Then n lines follow. ith line contains two integers w_i and c_i $(1 \le w_i \le 3,$ $1 \le c_i \le 10^9)$ — the weight and the cost of ith souvenir.

Output

Print one number — maximum possible total cost of souvenirs that Petya can carry to the market.

input
1 1 2 1

output
0

input
2 2 1 3 2 2

output
3

input
4 3 3 10 2 7 2 8 1 1

output
10

F. Card Game

2 seconds, 256 megabytes

Digital collectible card games have become very popular recently. So Vova decided to try one of these.

Vova has n cards in his collection. Each of these cards is characterised by its power p_i, magic number c_i and level l_i. Vova wants to build a deck with total power not less than k, but magic numbers may not allow him to do so — Vova can't place two cards in a deck if the sum of the magic numbers written on these cards is a prime number. Also Vova cannot use a card if its level is greater than the level of Vova's character.

At the moment Vova's character's level is 1. Help Vova to determine the minimum level he needs to reach in order to build a deck with the required total power.

Input

The first line contains two integers n and k ($1 \le n \le 100$, $1 \le k \le 100000$).

Then n lines follow, each of these lines contains three numbers that represent the corresponding card: p_i, c_i and l_i ($1 \le p_i \le 1000$, $1 \le c_i \le 100000$, $1 \le l_i \le n$).

Output

If Vova won't be able to build a deck with required power, print -1. Otherwise print the minimum level Vova has to reach in order to build a deck.

input
5 8 5 5 1 1 5 4 4 6 3 1 12 4 3 12 1

output
4

input
3 7 4 4 1 5 8 2 5 3 3

output
2

G. Anthem of Berland

3 seconds, 256 megabytes

Berland has a long and glorious history. To increase awareness about it among younger citizens, King of Berland decided to compose an anthem.

Though there are lots and lots of victories in history of Berland, there is the one that stand out the most. King wants to mention it in the anthem as many times as possible.

He has already composed major part of the anthem and now just needs to fill in some letters. King asked you to help him with this work.

The anthem is the string s of no more than 10^5 small Latin letters and question marks. The most glorious victory is the string t of no more than 10^5 small Latin letters. You should replace all the question marks with small Latin letters in such a way that the number of occurrences of string t in string s is maximal.

Note that the occurrences of string t in s can overlap. Check the third example for clarification.

Input

The first line contains string of small Latin letters and question marks s ($1 \leq |s| \leq 10^5$).

The second line contains string of small Latin letters t ($1 \leq |t| \leq 10^5$).

Product of lengths of strings $|s| \cdot |t|$ won't exceed 10^7.

Output

Output the maximum number of occurrences of string t you can achieve by replacing all the question marks in string s with small Latin letters.

input

```
winlose???winl???w??
win
```

output

```
5
```

input

```
glo?yto?e??an?
or
```

output

```
3
```

input

```
??c?????
abcab
```

output

```
2
```

In the first example the resulting string s is "winlose**win**winl**win**win"

In the second example the resulting string s is "glo**r**yto**r**e**or**an**d**". The last letter of the string can be arbitrary.

In the third example occurrences of string t are overlapping. String s with maximal number of occurrences of t is "**ab**c**abcab**".