A. Down the Hatch!

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Everybody knows that the Berland citizens are keen on health, especially students. Berland students are so tough that all they drink is orange juice!

Yesterday one student, Vasya and his mates made some barbecue and they drank this healthy drink only. After they ran out of the first barrel of juice, they decided to play a simple game. All n people who came to the barbecue sat in a circle (thus each person received a unique index b_i from 0 to $n - 1$). The person number 0 started the game (this time it was Vasya). All turns in the game were numbered by integers starting from 1. If the j-th turn was made by the person with index b_i, then this person acted like that:

1. he pointed at the person with index $(b_i + 1) \ mod \ n$ either with an elbow or with a nod ($x \ mod \ y$ is the remainder after dividing x by y);
2. if $j \geq 4$ and the players who had turns number $j - 1, j - 2, j - 3$, made during their turns the same moves as player b_i on the current turn, then he had drunk a glass of juice;
3. the turn went to person number $(b_i + 1) \ mod \ n$.

The person who was pointed on the last turn did not make any actions.

The problem was, Vasya's drunk too much juice and can't remember the goal of the game. However, Vasya's got the recorded sequence of all the participants' actions (including himself). Now Vasya wants to find out the maximum amount of juice he could drink if he played optimally well (the other players' actions do not change). Help him.

You can assume that in any scenario, there is enough juice for everybody.

Input

The first line contains a single integer n ($4 \leq n \leq 2000$) — the number of participants in the game. The second line describes the actual game: the i-th character of this line equals 'a', if the participant who moved i-th pointed at the next person with his elbow, and 'b', if the participant pointed with a nod. The game continued for at least 1 and at most 2000 turns.

Output

Print a single integer — the number of glasses of juice Vasya could have drunk if he had played optimally well.

Sample test(s)

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

input
```
4
abbab
```
output
```
0
```

Note

In both samples Vasya has got two turns — 1 and 5. In the first sample, Vasya could have drunk a glass of juice during the fifth turn if he had pointed at the next person with a nod. In this case, the sequence of moves would look like "abbbb". In the second sample Vasya wouldn't drink a single glass of juice as the moves performed during turns 3 and 4 are different.

B. Maximum Absurdity

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Reforms continue entering Berland. For example, during yesterday sitting the Berland Parliament approved as much as n laws (each law has been assigned a unique number from 1 to n). Today all these laws were put on the table of the President of Berland, G.W. Boosch, to be signed.

This time mr. Boosch plans to sign $2k$ laws. He decided to choose **exactly two** non-intersecting segments of integers from 1 to n of length k and sign all laws, whose numbers fall into these segments. More formally, mr. Boosch is going to choose two integers a, b $(1 \le a \le b \le n - k + 1, b - a \ge k)$ and sign all laws with numbers lying in the segments $[a; a + k - 1]$ and $[b; b + k - 1]$ (borders are included).

As mr. Boosch chooses the laws to sign, he of course considers the public opinion. Allberland Public Opinion Study Centre (APOSC) conducted opinion polls among the citizens, processed the results into a report and gave it to the president. The report contains the absurdity value for each law, in the public opinion. As mr. Boosch is a real patriot, he is keen on signing the laws with the maximum total absurdity. Help him.

Input

The first line contains two integers n and k $(2 \le n \le 2 \cdot 10^5, 0 < 2k \le n)$ — the number of laws accepted by the parliament and the length of one segment in the law list, correspondingly. The next line contains n integers $x_1, x_2, ..., x_n$ — the absurdity of each law $(1 \le x_i \le 10^9)$.

Output

Print two integers a, b — the beginning of segments that mr. Boosch should choose. That means that the president signs laws with numbers from segments $[a; a + k - 1]$ and $[b; b + k - 1]$. If there are multiple solutions, print the one with the minimum number a. If there still are multiple solutions, print the one with the minimum b.

Sample test(s)

input

```
5 2
3 6 1 1 6
```

output

```
1 4
```

input

```
6 2
1 1 1 1 1 1
```

output

```
1 3
```

Note

In the first sample mr. Boosch signs laws with numbers from segments [1;2] and [4;5]. The total absurdity of the signed laws equals $3 + 6 + 1 + 6 = 16$.

In the second sample mr. Boosch signs laws with numbers from segments [1;2] and [3;4]. The total absurdity of the signed laws equals $1 + 1 + 1 + 1 = 4$.

C. Students' Revenge

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A student's life is fraught with complications. Some Berland University students know this only too well. Having studied for two years, they contracted strong antipathy towards the chairperson of some department. Indeed, the person in question wasn't the kindest of ladies to begin with: prone to reforming groups, banning automatic passes and other mean deeds. At last the students decided that she just can't get away with all this anymore...

The students pulled some strings on the higher levels and learned that the next University directors' meeting is going to discuss n orders about the chairperson and accept exactly p of them. There are two values assigned to each order: a_i is the number of the chairperson's hairs that turn grey if she obeys the order and b_i — the displeasement of the directors if the order isn't obeyed. The students may make the directors pass any p orders chosen by them. The students know that the chairperson will obey exactly k out of these p orders. She will pick the orders to obey in the way that minimizes first, the directors' displeasement and second, the number of hairs on her head that turn grey.

The students want to choose p orders in the way that maximizes the number of hairs on the chairperson's head that turn grey. If there are multiple ways to accept the orders, then the students are keen on maximizing the directors' displeasement with the chairperson's actions. Help them.

Input

The first line contains three integers n ($1 \le n \le 10^5$), p ($1 \le p \le n$), k ($1 \le k \le p$) — the number of orders the directors are going to discuss, the number of orders to pass and the number of orders to be obeyed by the chairperson, correspondingly. Each of the following n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^9$), describing the corresponding order.

Output

Print **in an arbitrary order** p distinct integers — the numbers of the orders to accept so that the students could carry out the revenge. The orders are indexed from 1 to n in the order they occur in the input. If there are multiple solutions, you can print any of them.

Sample test(s)

input

```
5 3 2
5 6
5 8
1 3
4 3
4 11
```

output

```
3 1 2
```

input

```
5 3 3
10 18
18 17
10 20
20 18
20 18
```

output

```
2 4 5
```

Note

In the first sample one of optimal solutions is to pass orders 1, 2, 3. In this case the chairperson obeys orders number 1 and 2. She gets 10 new grey hairs in the head and the directors' displeasement will equal 3. Note that the same result can be achieved with order 4 instead of order 3.

In the second sample, the chairperson can obey all the orders, so the best strategy for the students is to pick the orders with the maximum sum of a_i values. The chairperson gets 58 new gray hairs and the directors' displeasement will equal 0.

D. Theft of Blueprints

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Insurgents accidentally got hold of the plan of a top secret research polygon created on a distant planet for the needs of the Galaxy Empire. The insurgents suppose that this polygon is developing new deadly weapon. The polygon consists of n missile silos connected by bidirectional underground passages. The passages are linked to laboratories where research is conducted. Naturally, the passages are guarded severely: the passage between silos i and j is patrolled by $c_{i,j}$ war droids.

The insurgents studied the polygon plan and noticed its unusual structure. As it turned out, for any k-element set of silos S there is exactly one silo that is directly connected by a passage with each silo from S (we'll call this silo *adjacent with* S). Having considered that, the insurgents decided to act as follows:

1. they choose a k-element set of silos S;
2. a group of scouts lands from the air into each silo from S;
3. each group moves along the corresponding passage to the silo, adjacent with S (as the scouts move, they check out the laboratories and watch for any signs of weapon blueprints);
4. in the silo, adjacent with S, the groups get on the ship and fly away.

The danger of the operation is the total number of droids that patrol the passages through which the scouts will go. The danger of the operation obviously only depends on the way to choose set S. The insurgents haven't yet decided on the exact silos to send the scouts to. However, they already want to start preparing the weapons for the scout groups. To do that, the insurgents need to know the mathematical average of the dangers of the operations that correspond to all possible ways to choose set S. Solve this problem to help the insurgents protect the ideals of the Republic!

Input

The first line contains two integers n and k ($2 \le n \le 2000$, $1 \le k \le n - 1$) — the number of silos and the number of scout groups, correspondingly. The next $n - 1$ lines describe the polygon plan: the i-th of these lines contains $n - i$ integers $c_{i,i+1}, c_{i,i+2}, \ldots, c_{i,n}$ — the number of droids that patrol the corresponding passages ($-1 \le c_{i,j} \le 10^9$; if $c_{i,j} = -1$, then silos i and j don't have a passage between them). All passages are bidirectional, that is, we can assume that $c_{i,j} = c_{j,i}$. No passages connect a silo with itself. It is guaranteed that the polygon plan meets the conditions of the problem statement.

Output

Print the average danger of the scouting operation, **rounded down to an integer**. Note that at the given limits the answer to the problem **always** fits into the standard integer 64-bit data type.

Please do not use the `%lld` specifier to write 64-bit integers in C++. It is preferred to use the `cout` stream or the `%I64d` specifier.

Sample test(s)

input

```
6 1
-1 -1 -1 8 -1
-1 5 -1 -1
-1 -1 3
-1 -1
-1
```

output

```
5
```

input

```
3 2
10 0
11
```

output

```
14
```

Note

In the first sample there are 6 one-element sets of silos. For sets $\{1\}$, $\{5\}$ the operation danger will equal 8, for sets $\{3\}$, $\{6\}$ — 3, for sets $\{2\}$, $\{4\}$ — 5. The mathematical average equals $\frac{8+8+3+3+5+5}{6} = \frac{16}{3}$.

In the second sample there are 3 two-elements sets of silos: $\{1,3\}$ (danger equals 21), $\{1,2\}$ (danger equals 11), $\{2,3\}$ (danger equals 10). The average operation danger equals $\frac{21+11+10}{3} = \frac{42}{3} = 14$.

E. Binary Key

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let's assume that p and q are strings of positive length, called the *container* and the *key* correspondingly, string q only consists of characters 0 and 1. Let's take a look at a simple algorithm that extracts *message* s from the given container p:

```
i = 0;
j = 0;
s = <>;
while i is less than the length of the string p
{
    if q[j] == 1, then add to the right of string s character p[i];
    increase variables i, j by one;
    if the value of the variable j equals the length of the string q, then j = 0;
}
```

In the given pseudocode i, j are integer variables, s is a string, '=' is an assignment operator, '==' is a comparison operation, '[]' is the operation of obtaining the string character with the preset index, '<>' is an empty string. We suppose that in all strings the characters are numbered starting from zero.

We understand that implementing such algorithm is quite easy, so your task is going to be slightly different. You need to construct the lexicographically minimum key of length k, such that when it is used, the algorithm given above extracts message s from container p (otherwise find out that such key doesn't exist).

Input

The first two lines of the input are non-empty strings p and s $(1 \le |p| \le 10^6,\ 1 \le |s| \le 200)$, describing the container and the message, correspondingly. The strings can contain any characters with the ASCII codes from 32 to 126, inclusive.

The third line contains a single integer k $(1 \le k \le 2000)$ — the key's length.

Output

Print the required key (string of length k, consisting only of characters 0 and 1). If the key doesn't exist, print the single character 0.

Sample test(s)

input

```
abacaba
aba
6
```

output

```
100001
```

input

```
abacaba
aba
3
```

output

```
0
```

Note

String $x = x_1x_2... x_p$ is *lexicographically smaller* than string $y = y_1y_2... y_q$, if either $p < q$ and $x_1 = y_1, x_2 = y_2, ... , x_p = y_p$, or there exists such integer r $(0 \le r < min(p, q))$ that $x_1 = y_1, x_2 = y_2, ... , x_r = y_r$ and $x_{r+1} < y_{r+1}$. Symbols are compared according to their ASCII codes.