VK Cup 2017 - Wild Card Round 1 (Unofficial Public Mirror)

A. Amusement Park

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

Pupils decided to go to amusement park. Some of them were with parents. In total, n people came to the park and they all want to get to the most extreme attraction and roll on it exactly **once**.

Tickets for group of x people are sold on the attraction, there should be at least one adult in each group (it is possible that the group consists of one adult). The ticket price for such group is $c_1 + c_2 \cdot (x - 1)^2$ (in particular, if the group consists of one person, then the price is c_1).

All pupils who came to the park and their parents decided to split into groups in such a way that each visitor join exactly one group, and the total price of visiting the most extreme attraction is as low as possible. You are to determine this minimum possible total price. There should be at least one adult in each group.

Input

The first line contains three integers n, c_1 and c_2 $(1 \le n \le 200\,000,\ 1 \le c_1, c_2 \le 10^7)$ — the number of visitors and parameters for determining the ticket prices for a group.

The second line contains the string of length n, which consists of zeros and ones. If the i-th symbol of the string is zero, then the i-th visitor is a pupil, otherwise the i-th person is an adult. It is guaranteed that there is at least one adult. It is possible that there are no pupils.

Output

Print the minimum price of visiting the most extreme attraction for all pupils and their parents. Each of them should roll on the attraction exactly once.

Examples

input
```
3 4 1
011
```
output
```
8
```

input
```
4 7 2
1101
```
output
```
18
```

Note

In the first test one group of three people should go to the attraction. Then they have to pay $4 + 1 * (3 - 1)^2 = 8$.

In the second test it is better to go to the attraction in two groups. The first group should consist of two adults (for example, the first and the second person), the second should consist of one pupil and one adult (the third and the fourth person). Then each group will have a size of two and for each the price of ticket is $7 + 2 * (2 - 1)^2 = 9$. Thus, the total price for two groups is 18.

B. Significant Cups

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

Stepan is a very experienced olympiad participant. He has n cups for Physics olympiads and m cups for Informatics olympiads. Each cup is characterized by two parameters — its significance c_i and width w_i.

Stepan decided to expose some of his cups on a shelf with width d in such a way, that:

- there is at least one Physics cup and at least one Informatics cup on the shelf,
- the total width of the exposed cups does not exceed d,
- from each subjects (Physics and Informatics) some of the most significant cups are exposed (i. e. if a cup for some subject with significance x is exposed, then all the cups for this subject with significance greater than x must be exposed too).

Your task is to determine the maximum possible total significance, which Stepan can get when he exposes cups on the shelf with width d, considering all the rules described above. The total significance is the sum of significances of all the exposed cups.

Input

The first line contains three integers n, m and d $(1 \le n, m \le 100\,000,\ 1 \le d \le 10^9)$ — the number of cups for Physics olympiads, the number of cups for Informatics olympiads and the width of the shelf.

Each of the following n lines contains two integers c_i and w_i $(1 \le c_i, w_i \le 10^9)$ — significance and width of the i-th cup for Physics olympiads.

Each of the following m lines contains two integers c_j and w_j $(1 \le c_j, w_j \le 10^9)$ — significance and width of the j-th cup for Informatics olympiads.

Output

Print the maximum possible total significance, which Stepan can get exposing cups on the shelf with width d, considering all the rules described in the statement.

If there is no way to expose cups on the shelf, then print 0.

Examples

input

```
3 1 8
4 2
5 5
4 2
3 2
```

output

```
8
```

input

```
4 3 12
3 4
2 4
3 5
3 4
3 5
5 2
3 4
```

output

```
11
```

input

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

output

```
0
```

Note

In the first example Stepan has only one Informatics cup which must be exposed on the shelf. Its significance equals 3 and width equals 2, so after Stepan exposes it, the width of free space on the shelf becomes equal to 6. Also, Stepan must expose the second Physics cup (which has width 5), because it is the most significant cup for Physics (its significance equals 5). After that Stepan can not expose more cups on the shelf, because there is no enough free space. Thus, the maximum total significance of exposed cups equals to 8.

C. Maximum Number

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Stepan has the newest electronic device with a display. Different digits can be shown on it. Each digit is shown on a seven-section indicator like it is shown on the picture below.

So, for example, to show the digit 3 on the display, 5 sections must be highlighted; and for the digit 6, 6 sections must be highlighted.

The battery of the newest device allows to highlight at most n sections on the display.

Stepan wants to know the maximum possible integer number which can be shown on the display of his newest device. Your task is to determine this number. Note that this number must not contain leading zeros. Assume that the size of the display is enough to show any integer.

Input

The first line contains the integer n $(2 \le n \le 100\,000)$ — the maximum number of sections which can be highlighted on the display.

Output

Print the maximum integer which can be shown on the display of Stepan's newest device.

Examples

input
```
2
```
output
```
1
```

input
```
3
```
output
```
7
```

D. Lie or Truth

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

Vasya has a sequence of cubes and exactly one integer is written on each cube. Vasya exhibited all his cubes in a row. So the sequence of numbers written on the cubes in the order from the left to the right equals to $a_1, a_2, ..., a_n$.

While Vasya was walking, his little brother Stepan played with Vasya's cubes and changed their order, so now the sequence of numbers written on the cubes became equal to $b_1, b_2, ..., b_n$.

Stepan said that he swapped only cubes which where on the positions between l and r, inclusive, and did not remove or add any other cubes (i. e. he said that he reordered cubes between positions l and r, inclusive, in some way).

Your task is to determine if it is possible that Stepan said the truth, or it is guaranteed that Stepan deceived his brother.

Input

The first line contains three integers n, l, r ($1 \le n \le 10^5$, $1 \le l \le r \le n$) — the number of Vasya's cubes and the positions told by Stepan.

The second line contains the sequence $a_1, a_2, ..., a_n$ ($1 \le a_i \le n$) — the sequence of integers written on cubes in the Vasya's order.

The third line contains the sequence $b_1, b_2, ..., b_n$ ($1 \le b_i \le n$) — the sequence of integers written on cubes after Stepan rearranged their order.

It is guaranteed that Stepan did not remove or add other cubes, he only rearranged Vasya's cubes.

Output

Print "LIE" (without quotes) if it is guaranteed that Stepan deceived his brother. In the other case, print "TRUTH" (without quotes).

Examples

input

```
5 2 4
3 4 2 3 1
3 2 3 4 1
```

output

```
TRUTH
```

input

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

output

```
LIE
```

input

```
4 2 4
1 1 1 1
1 1 1 1
```

output

```
TRUTH
```

Note

In the first example there is a situation when Stepan said the truth. Initially the sequence of integers on the cubes was equal to [3, 4, 2, 3, 1]. Stepan could at first swap cubes on positions 2 and 3 (after that the sequence of integers on cubes became equal to [3, 2, 4, 3, 1]), and then swap cubes in positions 3 and 4 (after that the sequence of integers on cubes became equal to [3, 2, 3, 4, 1]).

In the second example it is not possible that Stepan said truth because he said that he swapped cubes only between positions 1 and 2, but we can see that it is guaranteed that he changed the position of the cube which was on the position 3 at first. So it is guaranteed that Stepan deceived his brother.

In the third example for any values l and r there is a situation when Stepan said the truth.

E. Big Number and Remainder

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

Stepan has a very big positive integer.

Let's consider all cyclic shifts of Stepan's integer (if we look at his integer like at a string) which are also integers (i.e. they **do not have** leading zeros). Let's call such shifts as *good shifts*. For example, for the integer 10203 the good shifts are the integer itself 10203 and integers 20310 and 31020.

Stepan wants to know the minimum remainder of the division by the given number m among all good shifts. Your task is to determine the minimum remainder of the division by m.

Input

The first line contains the integer which Stepan has. The length of Stepan's integer is between 2 and $200\,000$ digits, inclusive. It is guaranteed that Stepan's integer does not contain leading zeros.

The second line contains the integer m $(2 \leq m \leq 10^8)$ — the number by which Stepan divides good shifts of his integer.

Output

Print the minimum remainder which Stepan can get if he divides all good shifts of his integer by the given number m.

Examples

input

```
521
3
```

output

```
2
```

input

```
1001
5
```

output

```
0
```

input

```
5678901234567890123456789
10000
```

output

```
123
```

Note

In the first example all good shifts of the integer 521 (good shifts are equal to 521, 215 and 152) has same remainder 2 when dividing by 3.

In the second example there are only two good shifts: the Stepan's integer itself and the shift by one position to the right. The integer itself is 1001 and the remainder after dividing it by 5 equals 1. The shift by one position to the right equals to 1100 and the remainder after dividing it by 5 equals 0, which is the minimum possible remainder.

F. Pens And Days Of Week

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

Stepan has n pens. Every day he uses them, and on the i-th day he uses the pen number i. On the $(n+1)$-th day again he uses the pen number 1, on the $(n+2)$-th — he uses the pen number 2 and so on.

On every working day (from Monday to Saturday, inclusive) Stepan spends exactly 1 milliliter of ink of the pen he uses that day. On Sunday Stepan has a day of rest, he does not stend the ink of the pen he uses that day.

Stepan knows the current volume of ink in each of his pens. Now it's the **Monday morning** and Stepan is going to use the pen **number 1** today. Your task is to determine which pen will run out of ink before all the rest (that is, there will be no ink left in it), if Stepan will use the pens according to the conditions described above.

Input

The first line contains the integer n $(1 \le n \le 50\,000)$ — the number of pens Stepan has.

The second line contains the sequence of integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$, where a_i is equal to the number of milliliters of ink which the pen number i currently has.

Output

Print the index of the pen which will run out of ink before all (it means that there will be no ink left in it), if Stepan will use pens according to the conditions described above.

Pens are numbered in the order they are given in input data. The numeration begins from one.

Note that the answer is always unambiguous, since several pens can not end at the same time.

Examples

input

```
3
3 3 3
```

output

```
2
```

input

```
5
5 4 5 4 4
```

output

```
5
```

Note

In the first test Stepan uses ink of pens as follows:

1. on the day number 1 (Monday) Stepan will use the pen number 1, after that there will be 2 milliliters of ink in it;
2. on the day number 2 (Tuesday) Stepan will use the pen number 2, after that there will be 2 milliliters of ink in it;
3. on the day number 3 (Wednesday) Stepan will use the pen number 3, after that there will be 2 milliliters of ink in it;
4. on the day number 4 (Thursday) Stepan will use the pen number 1, after that there will be 1 milliliters of ink in it;
5. on the day number 5 (Friday) Stepan will use the pen number 2, after that there will be 1 milliliters of ink in it;
6. on the day number 6 (Saturday) Stepan will use the pen number 3, after that there will be 1 milliliters of ink in it;
7. on the day number 7 (Sunday) Stepan will use the pen number 1, but it is a day of rest so he will not waste ink of this pen in it;
8. on the day number 8 (Monday) Stepan will use the pen number 2, after that this pen will run out of ink.

So, the first pen which will not have ink is the pen number 2.

G. Perfectionist Arkadiy

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Arkadiy has lots square photos with size $a \times a$. He wants to put some of them on a rectangular wall with size $h \times w$.

The photos which Arkadiy will put on the wall must form a rectangular grid and the distances between neighboring vertically and horizontally photos and also the distances between outside rows and columns of photos to the nearest bound of the wall must be equal to x, where x is some non-negative **real** number. Look on the picture below for better understanding of the statement.

Arkadiy haven't chosen yet how many photos he would put on the wall, however, he want to put at least one photo. Your task is to determine the **minimum** value of x which can be obtained after putting photos, or report that there is no way to put positive number of photos and satisfy all the constraints. Suppose that Arkadiy has enough photos to make any valid arrangement according to the constraints.

Note that Arkadiy wants to put at least one photo on the wall. The photos should not overlap, should completely lie inside the wall bounds and should have sides parallel to the wall sides.

Input

The first line contains three integers a, h and w ($1 \le a, h, w \le 10^9$) — the size of photos and the height and the width of the wall.

Output

Print one non-negative real number — the minimum value of x which can be obtained after putting the photos on the wall. The absolute or the relative error of the answer must not exceed 10^{-6}.

Print `-1` if there is no way to put positive number of photos and satisfy the constraints.

Examples

input

```
2 18 13
```

output

```
0.5
```

input

```
4 4 4
```

output

```
0
```

input

```
3 4 3
```

output

```
-1
```

Note

In the first example Arkadiy can put 7 rows of photos with 5 photos in each row, so the minimum value of x equals to 0.5.

In the second example Arkadiy can put only 1 photo which will take the whole wall, so the minimum value of x equals to 0.

In the third example there is no way to put positive number of photos and satisfy the constraints described in the statement, so the answer is `-1`.

H. Repairing Of String

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

Stepan had a favorite string s which consisted of the lowercase letters of the Latin alphabet.

After graduation, he decided to remember it, but it was a long time ago, so he can't now remember it. But Stepan remembers some information about the string, namely the sequence of integers $c_1, c_2, ..., c_n$, where n equals the length of the string s, and c_i equals the number of substrings in the string s with the length i, consisting of the **same** letters. The substring is a sequence of consecutive characters in the string s.

For example, if the Stepan's favorite string is equal to "`tttesst`", the sequence c looks like: $c = [7, 3, 1, 0, 0, 0, 0]$.

Stepan asks you to help to repair his favorite string s according to the given sequence $c_1, c_2, ..., c_n$.

Input

The first line contains the integer n ($1 \le n \le 2000$) — the length of the Stepan's favorite string.

The second line contains the sequence of integers $c_1, c_2, ..., c_n$ ($0 \le c_i \le 2000$), where c_i equals the number of substrings of the string s with the length i, consisting of the same letters.

It is guaranteed that the input data is such that the answer always exists.

Output

Print the repaired Stepan's favorite string. If there are several answers, it is allowed to print any of them. The string should contain only lowercase letters of the English alphabet.

Examples

input

```
6
6 3 1 0 0 0
```

output

```
kkrrrq
```

input

```
4
4 0 0 0
```

output

```
abcd
```

Note

In the first test Stepan's favorite string, for example, can be the string "`kkrrrq`", because it contains 6 substrings with the length 1, consisting of identical letters (they begin in positions 1, 2, 3, 4, 5 and 6), 3 substrings with the length 2, consisting of identical letters (they begin in positions 1, 3 and 4), and 1 substring with the length 3, consisting of identical letters (it begins in the position 3).

I. Composing Of String

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

Stepan has a set of n strings. Also, he has a favorite string s.

Stepan wants to do the following. He will take some strings of his set and write them down one after another. It is possible that he will take some strings more than once, and will not take some of them at all.

Your task is to determine the minimum number of strings in the set which Stepan needs to take and write so that the string s appears as a subsequence in the resulting written down string.

For example, in the string "`abcd`" strings "`ad`", "`acd`", "`abcd`" appear as subsequences, and strings "`ba`", "`abdc`" don't appear as subsequences.

Input

The first line contains the integer n $(1 \le n \le 50)$ — the number of strings in Stepan's set.

The next n lines contain n non-empty strings consisting of lowercase letters of the English alphabet. The length of each of these strings does not exceed 50 symbols. It is possible that some strings from Stepan's set are the same.

The next line contains the non-empty string s, consisting of lowercase letters of the English alphabet — Stepan's favorite string. The length of this string doesn't exceed 2500 symbols.

Output

Print the minimum number of strings which Stepan should take from the set and write them down one after another so that the string s appears as a subsequence in the resulting written down string. Each string from the set should be counted as many times as Stepan takes it from the set.

If the answer doesn't exsist, print `-1`.

Examples

input

```
3
a
aa
a
aaa
```

output

```
2
```

input

```
4
ab
aab
aa
bb
baaab
```

output

```
3
```

input

```
2
aaa
bbb
aaacbbb
```

output

```
-1
```

Note

In the first test, Stepan can take, for example, the third and the second strings from the set, write them down, and get exactly his favorite string.

In the second example Stepan can take, for example, the second, the third and again the second strings from the set and write them down. Then he will get a string "`aabaaaab`", in which his favorite string "`baaab`" is a subsequence.

In the third test Stepan can not get his favorite string, because it contains the letter "`c`", which is not presented in any of the strings in the set.

J. Stepan's Series

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

Well, the series which Stepan watched for a very long time, ended. In total, the series had n episodes. For each of them, Stepan remembers either that he definitely has watched it, or that he definitely hasn't watched it, or he is unsure, has he watched this episode or not.

Stepan's dissatisfaction is the **maximum** number of consecutive series that Stepan did not watch.

Your task is to determine according to Stepan's memories if his dissatisfaction could be exactly equal to k.

Input

The first line contains two integers n and k ($1 \le n \le 100$, $0 \le k \le n$) — the number of episodes in the series and the dissatisfaction which should be checked.

The second line contains the sequence which consists of n symbols "Y", "N" and "?". If the i-th symbol equals "Y", Stepan remembers that he has watched the episode number i. If the i-th symbol equals "N", Stepan remembers that he hasn't watched the epizode number i. If the i-th symbol equals "?", Stepan doesn't exactly remember if he has watched the episode number i or not.

Output

If Stepan's dissatisfaction can be exactly equal to k, then print "YES" (without qoutes). Otherwise print "NO" (without qoutes).

Examples

input

```
5 2
NYNNY
```

output

```
YES
```

input

```
6 1
????NN
```

output

```
NO
```

Note

In the first test Stepan remembers about all the episodes whether he has watched them or not. His dissatisfaction is 2, because he hasn't watch two episodes in a row — the episode number 3 and the episode number 4. The answer is "YES", because $k = 2$.

In the second test $k = 1$, Stepan's dissatisfaction is greater than or equal to 2 (because he remembers that he hasn't watch at least two episodes in a row — number 5 and number 6), even if he has watched the episodes from the first to the fourth, inclusive.

K. Stepan and Vowels

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Stepan likes to repeat vowel letters when he writes words. For example, instead of the word "`pobeda`" he can write "`pobeeeedaaaaa`".

Sergey does not like such behavior, so he wants to write a program to format the words written by Stepan. This program must combine all consecutive equal vowels to a single vowel. The vowel letters are "`a`", "`e`", "`i`", "`o`", "`u`" and "`y`".

There are exceptions: if letters "`e`" or "`o`" repeat in a row exactly 2 times, like in words "`feet`" and "`foot`", the program must skip them and do not transform in one vowel. For example, the word "`iiiimpleeemeentatiioon`" must be converted to the word "`implemeentatioon`".

Sergey is very busy and asks you to help him and write the required program.

Input

The first line contains the integer n $(1 \leq n \leq 100\,000)$ — the number of letters in the word written by Stepan.

The second line contains the string s which has length that equals to n and contains only lowercase English letters — the word written by Stepan.

Output

Print the single string — the word written by Stepan converted according to the rules described in the statement.

Examples

input
```
13
pobeeeedaaaaa
```
output
```
pobeda
```

input
```
22
iiiimpleeemeentatiioon
```
output
```
implemeentatioon
```

input
```
18
aeiouyaaeeiioouuyy
```
output
```
aeiouyaeeioouy
```

input
```
24
aaaoooiiiuuuyyyeeeggghhh
```
output
```
aoiuyeggghhh
```

L. Bars

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

Polycarp's workday lasts exactly n minutes. He loves chocolate bars and can eat one bar in one minute. Today Polycarp has k bars at the beginning of the workday.

In some minutes of the workday Polycarp has important things to do and in such minutes he is not able to eat a chocolate bar. In other minutes he can either eat or not eat one chocolate bar. It is guaranteed, that in the first and in the last minutes of the workday Polycarp has no important things to do and he will always eat bars in this minutes to gladden himself at the begining and at the end of the workday. Also it is guaranteed, that k is strictly greater than 1.

Your task is to determine such an order of eating chocolate bars that the maximum break time between eating bars is as minimum as possible.

Consider that Polycarp eats a bar in the minute x and the next bar in the minute y ($x < y$). Then the break time is equal to $y - x - 1$ minutes. It is not necessary for Polycarp to eat all bars he has.

Input

The first line contains two integers n and k ($2 \le n \le 200\,000$, $2 \le k \le n$) — the length of the workday in minutes and the number of chocolate bars, which Polycarp has in the beginning of the workday.

The second line contains the string with length n consisting of zeros and ones. If the i-th symbol in the string equals to zero, Polycarp has no important things to do in the minute i and he can eat a chocolate bar. In the other case, Polycarp is busy in the minute i and can not eat a chocolate bar. It is guaranteed, that the first and the last characters of the string are equal to zero, and Polycarp always eats chocolate bars in these minutes.

Output

Print the minimum possible break in minutes between eating chocolate bars.

Examples

input

```
3 3
010
```

output

```
1
```

input

```
8 3
01010110
```

output

```
3
```

Note

In the first example Polycarp can not eat the chocolate bar in the second minute, so the time of the break equals to one minute.

In the second example Polycarp will eat bars in the minutes 1 and 8 anyway, also he needs to eat the chocolate bar in the minute 5, so that the time of the maximum break will be equal to 3 minutes.