Lab Exam Questions

1)
A. Hongcow Learns the Cyclic Shift
time limit per test
2 seconds
memory limit per test
256 megabytes

Hongcow is learning to spell! One day, his teacher gives him a word that he needs to learn to spell. Being a dutiful student, he immediately learns how to spell the word.
Hongcow has decided to try to make new words from this one. He starts by taking the word he just learned how to spell, and moves the last character of the word to the beginning of the word. He calls this a cyclic shift. He can apply cyclic shift many times. For example, consecutively applying cyclic shift operation to the word "abracadabra" Hongcow will get words "aabracadabr", "raabracadab" and so on.
Hongcow is now wondering how many distinct words he can generate by doing the cyclic shift arbitrarily many times. The initial string is also counted.

Input
The first line of input will be a single string *s* ($1 \leq |s| \leq 50$), the word Hongcow initially learns how to spell. The string *s* consists only of lowercase English letters ('a'–'z').

Output
Output a single integer equal to the number of distinct strings that Hongcow can obtain by applying the cyclic shift arbitrarily many times to the given string.

Examples

Input
abcd

Output
4

Input
bbb

Output
1

Input
yzyz

Output
2

Note
For the first sample, the strings Hongcow can generate are "abcd", "dabc", "cdab", and "bcda".
For the second sample, no matter how many times Hongcow does the cyclic shift, Hongcow can only generate "bbb".
For the third sample, the two strings Hongcow can generate are "yzyz" and "zyzy".

---

2)
You have to paint N boards of length {A0, A1, A2, A3 … AN-1}. There are K painters available and you are also given how much time a painter takes to paint 1 unit of board. You have to get this job done as soon as possible under the constraints that **any painter will only paint contiguous sections of board.**

- 2 painters cannot share a board to paint. That is to say, a board cannot be painted partially by one painter, and partially by another.
- A painter will only paint contiguous boards. Which means a configuration where painter 1 paints board 1 and 3 but not 2 is invalid.

Return the ans
Input :
N : Number of boards
K : Number of painters
T : Time taken by painter to paint 1 unit of board
L : A List which will represent length of each board

Output:
return minimum time to paint all boards

Example
Input :
N : 2
K : 2
T : 5
L : [1, 10]

Output : 50

---

3)
finding number of trailing 0s in n!
0<=n<=10^9

Input
5

Output
1

---

4)
http://codeforces.com/group/ChiHCHUq6C/contest/100818/attachments/download/3890/20152016-acmicpc-southeastern-european-regional-programming-contest-seerc-2015-en.pdf
Problem I -> I for Idiot
Test cases with output here:
https://www.dropbox.com/sh/miwocs530bewbz8/AACKuq3kWGDBzeknosiwDliqa?dl=0

Time Limit: 0.5 seconds (C/C++)
Memory Limit: 4 megabytes

SEERC organizing committee decided to make this year's opening ceremony in unusual way – organize a parade of contestants on the city streets. This year ***N*** people (contestants, coaches and guests), that represent universities from participating countries, will go on the streets, trying to impress spectators by original costumes and loud songs. Each university is represented by a group of people and has a unique identifier (ID) that is carried by each person in the group. To make the parade well organised and entertaining, each group should be lined up in several rows, each consisting of ***K*** people.

Only one university was not able to line up according to the mentioned rule, and you need to find it's ID.

Input
The first line at input contains integer **N** and **K**, separated by a single space ($1 \le N \le 1\,000\,000$, $2 \le K \le 1$). Following **N** lines contain IDs $C_1, C_2, \ldots, C_n$ ($0 \le C_i \le 1\,000\,000\,000$, $1 \le i \le N$) of **N** people.

Output
The single line at output should contain one integer – the answer for the problem.

Sample input	Sample output
10 3 1 1 2 3 1 3 3 2 2 2	2

5.
Given a number n, we can divide it in only three parts n/2, n/3 and n/4 (we will consider only integer part). The task is to find the maximum sum we can make by dividing number in three parts recursively and summing up them together.

http://www.geeksforgeeks.org/recursively-break-number-3-parts-get-maximum-sum/

Final Lab Exam Questions

*1)http://codeforces.com/problemset/problem/231/C

You are given an array of n integers $A_1, A_2, ..., A_n$. Your task is to find a number that occurs the maximum number of times in this array i.e. a mode of the array.
However, before looking for such number, you are allowed to perform not more than k following operations — choose an element from the array and add 1 to it.
Your task is to find the maximum number of occurrences of some number in the array after performing no more than k allowed operations. If there are several such numbers, your task is to find the minimum one.

Input
The first line contains two integers n and k ($1 \le n \le 10^5$; $0 \le k \le 10^9$) — the number of elements in the array and the number of operations you are allowed to perform, correspondingly.
The third line contains a sequence of n integers $A_1, A_2, ..., A_n$ ($|a_i| \le 10^9$) — the initial array. The numbers in the lines are separated by single spaces.

Output
In a single line print two numbers — the maximum number of occurrences of some number in the array after at most k allowed operations are performed, and the minimum number that reaches the given maximum occurrences. Separate the printed numbers by whitespaces.

Examples

Input
5 3
6 3 4 0 2
Output
3 4
Input
3 4
5 5 5
Output
3 5
Input
5 3
3 1 2 2 1
Output
4 2

Note

In the first sample you can increase the second element of the array once and increase the fifth element of the array twice. Thus, we get sequence 6, 4, 4, 0, 4, where number 4 occurs 3 times.

In the second sample you don't need to perform a single operation or increase each element by one. If we do nothing, we get array 5, 5, 5, if we increase each by one, we get 6, 6, 6. In both cases the maximum number of occurrences equals 3. So we should do nothing, as number 5 is less than number 6.

In the third sample we should increase the second array element once and the fifth element once. Thus, we get sequence 3, 2, 2, 2, 2, where number 2 occurs 4 times.

***2)http://codeforces.com/problemset/problem/182/D**

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

Let us define divisor for a string in the following way. String *a* is the divisor of string *b* if and only if there exists a positive integer *x* such that if we write out string *a* consecutively *x* times, we get string *b*. For example, string "abab" has two divisors — "ab" and "abab".
Write a program that calculates the number of **common divisors** of two strings and prints them.

Input
The first input line contains a non-empty string S_1.
The second input line contains a non-empty string S_2.
Lengths of strings S_1 and S_2 are do not exceed 10^4. The strings only consist of lowercase Latin letters.

Output
In the first line print the number of common divisors of S_1 and S_2.
In the next line print the common divisors of strings S_1 and S_2 in increasing order of their length.

Examples
Input
abcdabcd
abcdabcdabcdabcd

Output
2
abcdabcd
abcdabcdabcdabcd

Input
aaa
aa

Output
1
a

Note
In first sample the common divisors are strings "abcd" and "abcdabcd".
In the second sample the common divisor is a single string "a". String "aa" isn't included in the answer as it isn't a divisor of string "aaa".

---

3)http://codeforces.com/problemset/problem/248/B

time limit per test

2 seconds

memory limit per test

256 megabytes

input

standard input

output

standard output

Chilly Willy loves playing with numbers. He only knows prime numbers that are digits yet. These numbers are 2, 3, 5 and 7. But Willy grew rather bored of such numbers, so he came up with a few games that were connected with them.

Chilly Willy wants to find the minimum number of length *n*, such that it is simultaneously divisible by all numbers Willy already knows (2, 3, 5 and 7). Help him with that.

A number's length is the number of digits in its decimal representation without leading zeros.

Input

A single input line contains a single integer *n* ($1 \leq n \leq 10^5$).

Output

Print a single integer — the answer to the problem without leading zeroes, or "-1" (without the quotes), if the number that meet the problem condition does not exist.

Examples

Input

1

Output

-1

Input

5

Output

10080

***4)http://codeforces.com/problemset/problem/430/B**

time limit per test

1 second

memory limit per test

256 megabytes

input

standard input

output

standard output

Pepe is training for the IOI. What is a better way to train than playing a Zuma-like game?

There are n balls put in a row. Each ball is colored in one of k colors. Initially the row doesn't contain three or more contiguous balls with the same color. Pepe has a single ball of color x. He can insert his ball at any position in the row (probably, between two other balls). If at any moment there are three or more contiguous balls of the same color in the row, they are destroyed immediately. This rule is applied multiple times, until there are no more sets of 3 or more contiguous balls of the same color.

For example, if Pepe has the row of balls [black, black, white, white, black, black] and a white ball, he can insert the ball between two white balls. Thus three white balls are destroyed, and then four black balls become contiguous, so all four balls are destroyed. The row will not contain any ball in the end, so Pepe can destroy all 6 balls.

Pepe wants to destroy as many balls as possible. You are given the description of the row of balls, and the color of Pepe's ball. Help Pepe by telling him the maximum number of balls he can destroy.

Input

The first line of input contains three integers: n ($1 \le n \le 1000$), k ($1 \le k \le 100$) and x ($1 \le x \le k$). The next line contains n space-separated integers $c_1, c_2, ..., c_n$ ($1 \le c_i \le k$). Number c_i means that the i-th ball in the row has color c_i.

It is guaranteed that the initial row of balls will never contain three or more contiguous balls of the same color.

Output

Print a single integer — the maximum number of balls Pepe can destroy.

Examples

Input

6 2 2
1 1 2 2 1 1

Output

6

Input

1 1 1
1

Output

0

***5)http://codeforces.com/problemset/problem/626/B**

time limit per test

2 seconds

memory limit per test
256 megabytes
input
standard input
output
standard output
You are given a deck of n cards, each of which is either red, green, or blue. As long as there are at least two cards left, you can do one of two actions:

- take any two (not necessarily adjacent) cards with different colors and exchange them for a new card of the third color;
- take any two (not necessarily adjacent) cards with the same color and exchange them for a new card with that color.

You can repeats this process until there is only one card left. What are the possible colors for the final card?

Input

The first line of the input contains a single integer n ($1 \le n \le 200$) — the total number of cards.

The next line contains a string s of length n — the colors of the cards. s contains only the characters 'B', 'G', and 'R', representing blue, green, and red, respectively.

Output

Print a single string of up to three characters — the possible colors of the final card (using the same symbols as the input) in alphabetical order.

Examples

Input
2
RB
Output
G
Input
3
GRG
Output
BR
Input
5
BBBBB
Output
B

Note

In the first sample, you one red card and one blue card, which you can exchange for a green card.

In the second sample, you have two green cards and one red card. You have two options: exchange the two green cards for a green card, then exchange the new green card and the red card for a blue card. Alternatively, you can exchange a green and a red card for a blue card, then exchange the blue card and remaining green card for a red card.
In the third sample, you only have blue cards, so you can only exchange them for more blue cards.

---

6)http://codeforces.com/problemset/problem/607/A
time limit per test
2 seconds
memory limit per test
256 megabytes
input
standard input
output
standard output

There are n beacons located at distinct positions on a number line. The i-th beacon has position a_i and power level b_i. When the i-th beacon is activated, it destroys all beacons to its left (direction of decreasing coordinates) within distance b_i inclusive. The beacon itself is not destroyed however. Saitama will activate the beacons one at a time from right to left. If a beacon is destroyed, it cannot be activated.
Saitama wants Genos to add a beacon strictly to the right of all the existing beacons, with any position and any power level, such that the least possible number of beacons are destroyed. Note that Genos's placement of the beacon means it will be the first beacon activated. Help Genos by finding the minimum number of beacons that could be destroyed.

Input
The first line of input contains a single integer n ($1 \le n \le 100\ 000$) — the initial number of beacons.
The i-th of next n lines contains two integers a_i and b_i ($0 \le a_i \le 1\ 000\ 000$, $1 \le b_i \le 1\ 000\ 000$) — the position and power level of the i-th beacon respectively. No two beacons will have the same position, so $a_i \neq a_j$ if $i \neq j$.

Output
Print a single integer — the minimum number of beacons that could be destroyed if exactly one beacon is added.

Examples
Input
4
1 9
3 1
6 1
7 4
Output
1

Input
7
1 1
2 1
3 1
4 1
5 1
6 1
7 1
Output
3

---

7)
A function you wrote to insert elements in a linked has a bug and is doing unexpected things. Irrespective of the number you want to insert into the linked list, it is only inserting either a 1 or a 10.

- If the sum of elements in the linked list is even
    - It inserts 1 with a probability of *a*
    - Or insert 10 with a probability of 1 - *a*
        - At the end of the linked list with probability *b*
        - At the start with probability 1 - *b*
- Else if the sum is odd,
    - It inserts 10 with a probability of *c*
    - Or insert 1 with a probability of 1 - *c*
        - At the start of the linked list with probability *d*
        - At the end with probability 1 - *d*

Given the initial linked list of size *n* and the number of insert operations *k* find the expected value of the sum of elements in the linked list after these insert operations.

1 <= *n* <= 100000

0 <= *a, b, c, d* <= 1
1 <= *k* <= 10^9

Sample
Input
n = 3
k = 1
a = *b* = *c* = *d* = 0.5
LL = 3, 6, 7
Output
21.5

***8)Linked List**
time limit per test
2 seconds
memory limit per test
256 megabytes
input
standard input
output
standard output

In this question you have to write a function which takes a linked list as an input and returns the modified version of the linked list. You should **NOT** make any changes to the main function or the struct. Only make changes inside the required function.
Given a singly linked list and an integer *K*, reverses the nodes of the list *K* nodes at a time and returns modified linked list.

Function Arguments
Pointer to the head of the linked list and value of K. It is guaranteed that the linked list has no loops and has a tail whose next is NULL.

Return Value
Return the pointer to the new head.
NO NEW NODES MUST BE CREATED.
You must only modify the existing linked list to make the required changes.
DONOT USE MALLOC. If found, you will be awarded zero marks.

Constraints
N = Number of nodes in the linked list <= 10^5

K is a divisor of N

Base Code

```
/**
 * Definition for singly-linked list. DONOT modify or uncomment this.
 * typedef struct ListNode{
 *        int val;
 *        struct ListNode *next;
 * }ListNode;
 */

ListNode* reverseList(ListNode* L, int K) {
        /* Your code here */
}
```

Examples

Input
1 -> 2 -> 3 -> 4 -> 5 -> 6, K = 2
Output
2 -> 1 -> 4 -> 3 -> 6 -> 5

Input
5->6->8->7->13, K = 5
Output
13->7->8->6->5

Alternate Base Code

```
#include<stdio.h>
#include<stdlib.h>

//Definition for singly-linked list.
typedef struct ListNode{
        //DONOT modify the struct.
        int val;
        struct ListNode *next;
}ListNode;

ListNode* reverseList(ListNode* L, int K) {
        /* Your code here */
}
```

```
int main(){
	// DONOT make any changes to the main function.
	ListNode *head = NULL, *tmp;
	void **ar1, **ar2;
	int n, k, i, j;
	scanf("%d %d", &n, &k);
	if(n%k != 0){printf("K must divide N\n");}
	ar1 = (void **)malloc(sizeof(void *)*n);
	ar2 = (void **)malloc(sizeof(void *)*n);

	for(i = 0;i < n;i++){
		int x;
		tmp = head;
		head = (ListNode *)malloc(sizeof(ListNode));
		scanf("%d", &x);
		head->val = x;
		head->next = tmp;
	}

	tmp = head;
	i = 0;
	while(tmp != NULL){
		ar1[i++] = tmp;
		tmp = tmp->next;
	}
	tmp = reverseList(head, k);
	i = 0;
	while(tmp != NULL){
		ar2[i++] = tmp;
		tmp = tmp->next;
	}
	for(i = 0;i < n;i += k){
		for(j = 0;j < k;j++){
			printf("%p ", ar1[i + j] - ar2[i + k - 1 - j]);
		}
	}

	return 0;
}
```

Alternate Examples

Input

6 2
1 2 3 4 5 6
Output
2 1 4 3 6 5
(nil) (nil) (nil) (nil) (nil) (nil)

Input
5 5
5 6 8 7 13
Output
13 7 8 6 5
(nil) (nil) (nil) (nil) (nil)