Codeforces Round #544 (Div. 3)

A. Middle of the Contest

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

Polycarp is going to participate in the contest. It starts at $h_1 : m_1$ and ends at $h_2 : m_2$. It is guaranteed that the contest lasts an even number of minutes (i.e. $m_1 \% 2 = m_2 \% 2$, where $x \% y$ is x modulo y). It is also guaranteed that the entire contest is held during a single day. And finally it is guaranteed that the contest lasts at least two minutes.

Polycarp wants to know the time of the midpoint of the contest. For example, if the contest lasts from $10 : 00$ to $11 : 00$ then the answer is $10 : 30$, if the contest lasts from $11 : 10$ to $11 : 12$ then the answer is $11 : 11$.

Input

The first line of the input contains two integers h_1 and m_1 in the format hh:mm.

The second line of the input contains two integers h_2 and m_2 in the same format (hh:mm).

It is guaranteed that $0 \le h_1, h_2 \le 23$ and $0 \le m_1, m_2 \le 59$.

It is guaranteed that the contest lasts an even number of minutes (i.e. $m_1 \% 2 = m_2 \% 2$, where $x \% y$ is x modulo y). It is also guaranteed that the entire contest is held during a single day. And finally it is guaranteed that the contest lasts at least two minutes.

Output

Print two integers h_3 and m_3 ($0 \le h_3 \le 23, 0 \le m_3 \le 59$) corresponding to the midpoint of the contest in the format hh:mm. Print each number as exactly two digits (prepend a number with leading zero if needed), separate them with ':'.

Examples

input
10:00 11:00
output
10:30

input
11:10 11:12
output
11:11

input
01:02 03:02
output
02:02

B. Preparation for International Women's Day

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

International Women's Day is coming soon! Polycarp is preparing for the holiday.

There are n candy boxes in the shop for sale. The i-th box contains d_i candies.

Polycarp wants to prepare the maximum number of gifts for k girls. Each gift will consist of **exactly two** boxes. The girls should be able to share each gift equally, so the total amount of candies in a gift (in a pair of boxes) should be divisible by k. In other words, two boxes i and j ($i \neq j$) can be combined as a gift if $d_i + d_j$ is divisible by k.

How many boxes will Polycarp be able to give? Of course, each box can be a part of no more than one gift. Polycarp cannot use boxes "partially" or redistribute candies between them.

Input
The first line of the input contains two integers n and k ($1 \le n \le 2 \cdot 10^5, 1 \le k \le 100$) — the number the boxes and the number the girls.

The second line of the input contains n integers $d_1, d_2, \dots, d_n$ ($1 \le d_i \le 10^9$), where d_i is the number of candies in the i-th box.

Output
Print one integer — the maximum number of the boxes Polycarp can give as gifts.

Examples

input
7 2 1 2 2 3 2 4 10
output
6

input
8 2 1 2 2 3 2 4 6 10
output
8

input
7 3 1 2 2 3 2 4 5
output
4

Note
In the first example Polycarp can give the following pairs of boxes (pairs are presented by **indices of corresponding boxes**):

- $(2, 3)$;
- $(5, 6)$;
- $(1, 4)$.

So the answer is 6.

In the second example Polycarp can give the following pairs of boxes (pairs are presented by **indices of corresponding boxes**):

- $(6, 8)$;
- $(2, 3)$;
- $(1, 4)$;
- $(5, 7)$.

So the answer is 8.

In the third example Polycarp can give the following pairs of boxes (pairs are presented by **indices of corresponding boxes**):

- $(1, 2)$;
- $(6, 7)$.

So the answer is 4.

C. Balanced Team

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

You are a coach at your local university. There are n students under your supervision, the programming skill of the i-th student is a_i.

You have to create a team for a new programming competition. As you know, the more students some team has the more probable its victory is! So you have to create a team with the maximum number of students. But you also know that a team should be *balanced*. It means that the programming skill of each pair of students in a created team should differ by no more than 5.

Your task is to report the maximum possible number of students in a *balanced* team.

Input
The first line of the input contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of students.

The second line of the input contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$), where a_i is a programming skill of the i-th student.

Output
Print one integer — the maximum possible number of students in a *balanced* team.

Examples

input
6 1 10 17 12 15 2
output
3

input
10 1337 1337 1337 1337 1337 1337 1337 1337 1337 1337
output
10

input
6 1 1000 10000 10 100 1000000000
output
1

Note
In the first example you can create a team with skills $[12, 17, 15]$.

In the second example you can take all students in a team because their programming skills are equal.

In the third example you can create a team consisting of a single student (and you cannot create a team consisting of at least two students).

D. Zero Quantity Maximization

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

You are given two arrays a and b, each contains n integers.

You want to create a new array c as follows: choose some real (i.e. not necessarily integer) number d, and then for every $i \in [1, n]$ let $c_i := d \cdot a_i + b_i$.

Your goal is to maximize the number of zeroes in array c. What is the largest possible answer, if you choose d optimally?

Input
The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of elements in both arrays.

The second line contains n integers a_1, a_2, ..., a_n ($-10^9 \le a_i \le 10^9$).

The third line contains n integers b_1, b_2, ..., b_n ($-10^9 \le b_i \le 10^9$).

Output
Print one integer — the maximum number of zeroes in array c, if you choose d optimally.

Examples

input
5 1 2 3 4 5 2 4 7 11 3
output
2

input
3 13 37 39 1 2 3
output
2

input

4 0 0 0 0 1 2 3 4
output
0

input
3 1 2 -1 -6 -12 6
output
3

Note

In the first example, we may choose $d = -2$.

In the second example, we may choose $d = -\frac{1}{13}$.

In the third example, we cannot obtain any zero in array c, no matter which d we choose.

In the fourth example, we may choose $d = 6$.

E. K Balanced Teams

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

You are a coach at your local university. There are n students under your supervision, the programming skill of the i-th student is a_i.

You have to form k teams for yet another new programming competition. As you know, the more students are involved in competition the more probable the victory of your university is! So you have to form no more than k (and at least one) **non-empty** teams so that the **total** number of students in them is maximized. But you also know that **each** team should be *balanced*. It means that the programming skill of each pair of students in **each** team should differ by no more than 5. Teams are independent from one another (it means that the difference between programming skills of two students from two different teams does not matter).

It is possible that some students not be included in any team at all.

Your task is to report the maximum possible **total** number of students in no more than k (and at least one) **non-empty** *balanced* teams.

If you are Python programmer, consider using PyPy instead of Python when you submit your code.

Input

The first line of the input contains two integers n and k ($1 \le k \le n \le 5000$) — the number of students and the maximum number of teams, correspondingly.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), where a_i is a programming skill of the i-th student.

Output

Print one integer — the maximum possible **total** number of students in no more than k (and at least one) **non-empty** *balanced* teams.

Examples

input
5 2 1 2 15 15 15
output
5

input
6 1 36 4 1 25 9 16
output
2

input
4 4 1 10 100 1000
output
4

F1. Spanning Tree with Maximum Degree

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

You are given an undirected unweighted connected graph consisting of n vertices and m edges. It is guaranteed that there are no self-loops or multiple edges in the given graph.

Your task is to find **any** spanning tree of this graph such that the maximum degree over all vertices is maximum possible. Recall that the degree of a vertex is the number of edges incident to it.

Input

The first line contains two integers n and m ($2 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le min(2 \cdot 10^5, \frac{n(n-1)}{2})$) — the number of vertices and edges, respectively.

The following m lines denote edges: edge i is represented by a pair of integers v_i, u_i ($1 \le v_i, u_i \le n$, $u_i \ne v_i$), which are the indices of vertices connected by the edge. There are no loops or multiple edges in the given graph, i. e. for each pair (v_i, u_i) there are no other pairs (v_i, u_i) or (u_i, v_i) in the list of edges, and for each pair (v_i, u_i) the condition $v_i \ne u_i$ is satisfied.

Output

Print $n - 1$ lines describing the edges of a spanning tree such that the maximum degree over all vertices is maximum possible. Make sure that the edges of the printed spanning tree form some subset of the input edges (order doesn't matter and edge (v, u) is considered the same as the edge (u, v)).

If there are multiple possible answers, print any of them.

Examples

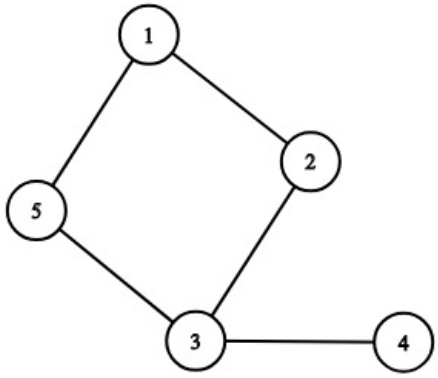
input
5 5 1 2 2 3 3 5 4 3 1 5
output
3 5 2 1 3 2 3 4

input
4 6 1 2 1 3 1 4 2 3 2 4 3 4
output
4 1 1 2 1 3

input
8 9 1 2 2 3 2 5 1 6 3 4 6 5 4 5 2 7 5 8
output
3 2 2 5 8 5 6 1 2 7 1 2 3 4

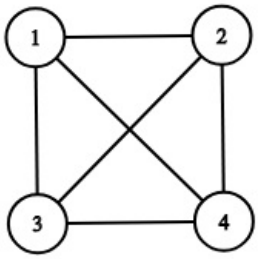
Note

Picture corresponding to the first example:



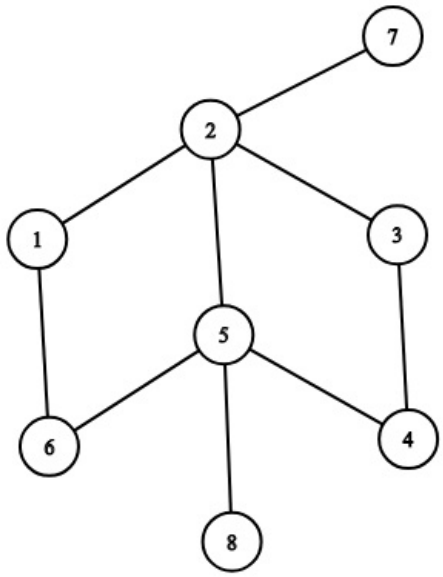
In this example the number of edges of spanning tree incident to the vertex 3 is 3. It is the maximum degree over all vertices of the spanning tree. It is easy to see that we cannot obtain a better answer.

Picture corresponding to the second example:



In this example the number of edges of spanning tree incident to the vertex 1 is 3. It is the maximum degree over all vertices of the spanning tree. It is easy to see that we cannot obtain a better answer.

Picture corresponding to the third example:



In this example the number of edges of spanning tree incident to the vertex 2 is 4. It is the maximum degree over all vertices of the spanning tree. It is easy to see that we cannot obtain a better answer. But because this example is symmetric, we can choose almost the same spanning tree but with vertex 5 instead of 2.

F2. Spanning Tree with One Fixed Degree

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

You are given an undirected unweighted connected graph consisting of n vertices and m edges. It is guaranteed that there are no self-loops or multiple edges in the given graph.

Your task is to find **any** spanning tree of this graph such that the **degree of the first vertex (vertex with label 1 on it)** is equal to D (or say that there are no such spanning trees). Recall that the degree of a vertex is the number of edges incident to it.

Input

The first line contains three integers n, m and D ($2 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le min(2 \cdot 10^5, \frac{n(n-1)}{2})$, $1 \le D < n$) — the number of vertices, the number of edges and required degree of the first vertex, respectively.

The following m lines denote edges: edge i is represented by a pair of integers v_i, u_i ($1 \le v_i, u_i \le n$, $u_i \ne v_i$), which are the indices of vertices connected by the edge. There are no loops or multiple edges in the given graph, i. e. for each pair (v_i, u_i) there are no other pairs (v_i, u_i) or (u_i, v_i) in the list of edges, and for each pair (v_i, u_i) the condition $v_i \ne u_i$ is satisfied.

Output

If there is no spanning tree satisfying the condition from the problem statement, print "NO" in the first line.

Otherwise print "YES" in the first line and then print $n - 1$ lines describing the edges of a spanning tree such that the **degree of the first vertex (vertex with label 1 on it)** is equal to D. Make sure that the edges of the printed spanning tree form some subset of the input edges (order doesn't matter and edge (v, u) is considered the same as the edge (u, v)).

If there are multiple possible answers, print any of them.

Examples

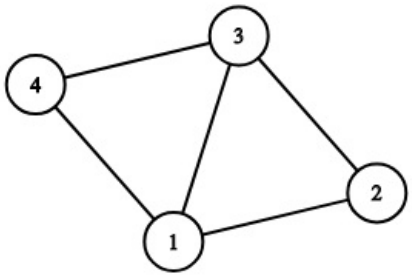
input
4 5 1 1 2 1 3 1 4 2 3 3 4
output
YES 2 1 2 3 3 4

input
4 5 3 1 2 1 3 1 4 2 3 3 4
output
YES 1 2 1 3 4 1

input
4 4 3 1 2 1 4 2 3 3 4
output
NO

Note

The picture corresponding to the first and second examples:



The picture corresponding to the third example: