Problem A. Air Flights

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Air traffic was significantly different in 2021 from all previous years, so you were instructed to collect statistics on completed SWIM Airlines flights.

SWIM Airlines used a fixed schedule in 2021: formally, depending on the day of the week, there was completed the same number of flights throughout the year. Every Monday there were a_1 flights, every Tuesday there were a_2 flights, and so on until Sunday, when a_7 flights were conducted on each Sunday.

According to the provided schedule calculate the number of flights SWIM Airlines has made in 2021.

For your information, 2021 is not a leap year, and the 1st of January is a Friday.

Input

Each of 7 input lines contains a single integer: $a_1, \ldots, a_7$ ($0 \le a_i \le 100$), the number of flights for each day of the week from Monday to Sunday.

Output

Print a single integer, the total number of flights in 2021.

Examples

standard input	standard output
1 0 0 0 0 0 0	52
0 1 0 0 0 0 0	52
0 0 1 0 0 0 0	52
0 0 0 1 0 0 0	52
0 0 0 0 0 1 0	52

Note

Following examples of source codes are the starting templates for most popular programming languages of the competition.

Note, a submission of these examples will get some points for the problem above.

Python

```
a1 = int(input())
a2 = int(input())
a3 = int(input())
a4 = int(input())
a5 = int(input())
a6 = int(input())
```

```
a7 = int(input())

result = 52

print(result)
```

C++

```
#include <iostream>

using namespace std;

int main() {
    int a1, a2, a3, a4, a5, a6, a7;
    cin >> a1;
    cin >> a2;
    cin >> a3;
    cin >> a4;
    cin >> a5;
    cin >> a6;
    cin >> a7;

    long long result = 52;
    cout << result << endl;
    return 0;
}
```

C

```
#include <stdio.h>

int main() {
    int a1, a2, a3, a4, a5, a6, a7;
    scanf("%d", &a1);
    scanf("%d", &a2);
    scanf("%d", &a3);
    scanf("%d", &a4);
    scanf("%d", &a5);
    scanf("%d", &a6);
    scanf("%d", &a7);

    int result = 52;

    printf("%d", result);
    return 0;
}
```

Java

```
import java.util.Scanner;

public class Main {
    public static void main(String[] arg) {
        Scanner in = new Scanner(System.in);
        int a1 = in.nextInt();
        int a2 = in.nextInt();
```

```
        int a3 = in.nextInt();
        int a4 = in.nextInt();
        int a5 = in.nextInt();
        int a6 = in.nextInt();
        int a7 = in.nextInt();

        int result = 52;

        System.out.println(result);
    }
}
```

Scoring

There are 100 tests in this task, each test is scored at exactly one point. Thus, the number of points received for this task is equal to the number of passed tests. Sample tests from the statement are also included in the 100 evaluated tests.

Problem B. Blooming Sakura

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Eugene is going to Japan for his summer vacation. One warm July evening he was walking along the line of blooming sakura trees.

Sakura is a rooted tree with n vertices indexed from 1 to n. The vertex u is called the descendant of the vertex v if $v \neq u$ and v is on the unique path from u to the root of the tree. Let us denote the number of descendants of v as a_v. The sakura is called blooming if the sum of a_i for all i from 1 to n is equal to k.

You must find out if there is at least one blooming sakura, and if it exists, you might be asked to print its description.

Input

You must input three integers, one on a line: n, k, t $(1 \le n \le 10^6, 0 \le k \le 10^9, 0 \le t \le 1)$ — the number of vertices of sakura, the sum of descendant number for all vertices of the blooming sakura, and the indicator if you should print the sakura description. $t = 0$ means that you must not output the sakura description, $t = 1$ — means that you must output it.

The root of the tree is vertex number 1.

Output

If the blooming sakura doesn't exist, output the word "NO". If the blooming sakura exists, output the word "YES". If $t = 1$, the following line must contain the description of the blooming sakura: output $n-1$ integers — indices of parents of vertices $2, 3, \ldots, n$, respectively.

Examples

standard input	standard output
4 7 0	NO
6 9 1	YES 1 2 3 2 1

Scoring

There are six groups of tests. Points for each group are awarded only if all tests of the group and all tests of the **required** groups have successfully passed.

Group	Points	Additional constraints			Req. groups	Comment
		n	k	t		
0	0	–	–	–	–	Sample tests
1	7	$n \leq 8$	–	–	0	
2	23	$n \leq 1500$	–	–	0 – 1	
3	11	–	$k = n$	–	–	
4	13	–	$k = \frac{n(n-1)}{2}$	–	–	
5	17	–	–	$t = 0$	–	
6	29	–	–	–	0 – 5	

Problem C. n-dimensional chocolate

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

Mom bought her little Vasya an n-dimensional chocolate. The size of the chocolate's i-th dimension equals to a_i millimeters. Vasya wants to share the chocolate with his friends so he plans to cut it in exactly k pieces.

As all regular chocolates do, this chocolate consists of n-dimensional cubic slices 1 millimeter in size on all dimensions. One can cut the chocolate only between slices. Moreover, each cut should pass through the whole chocolate even if it was cut already. Finally, each cut should be parallel to all dimensions except for one. Only when all cuts are done Vasya disassembles his chocolate in pieces. In other words, if Vasya makes b_i cuts perpendicular to the i-th dimension ($b_i \le a_i - 1$), the chocolate will be split into $(b_1 + 1) \cdot (b_2 + 1) \cdot \ldots \cdot (b_n + 1)$ pieces.

Vasya wants to cut the chocolate in a way that he thinks fair, that is to maximize the size of the smallest of the resulting pieces. The size of a piece equals to the number of n-dimensional slices in it.

As Mom cares that Vasya doesn't eat too much sweets, the number of slices in the chocolate before all cuts doesn't exceed 10^{18}.

If there is no way to cut the given chocolate into exactly k piece, print `0`.

Input

The first line contains two integers n and k ($1 \le n \le 100\,000, 1 \le k \le 10^{16}$) — the number of dimensions and the number of pieces Vasya wants to obtain.

The following line contains n positive integers a_i, the i-th of them equals to the size of the chocolate's i-th dimension. It is guaranteed that $a_1 \cdot a_2 \cdot \ldots \cdot a_n \le 10^{18}$

Output

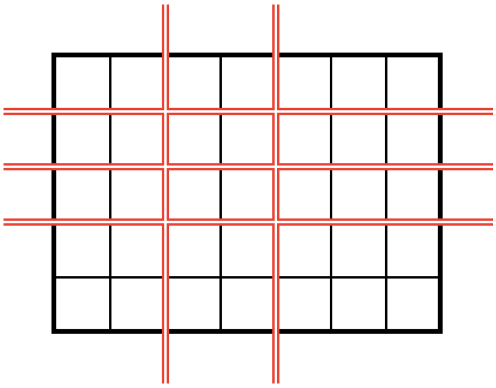
Print one integer — maximum possible size of the smallest piece if the chocolate is split into exactly k pieces. If there is no way to split the chocolate into exactly k pieces, print `0`.

Examples

standard input	standard output
`2 12` `5 7`	`2`
`3 2` `3 5 7`	`45`
`3 10` `3 6 9`	`12`
`2 11` `9 10`	`0`

Note

In the first test it is impossible to split the chocolate into exactly 12 piece of size greater than 2.



Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **required** groups. Note that for some groups it is not required to pass sample tests.

Group	Points	Additional constraints		Req. groups	Comment
		n	k		
0	0	–	–	–	Sample tests.
1	7	$n \leq 100$	$k \leq 3$		
2	17	$n \leq 100$	$k \leq 1000$	0, 1	
3	13	$n \leq 100\,000$	$k \leq 10\,000$	0, 1, 2	
4	15	$n \leq 100\,000$	$k \leq 10^{10}$	0, 1, 2, 3	
5	16	$n \leq 3$	$k \leq 10^{16}$	0	
6	32	$n \leq 100\,000$	$k \leq 10^{16}$	0, 1, 2, 3, 4, 5	

Problem D. Journey planning

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

You are planning a journey to Moscow with your friend. You have already bought tickets, so you know that you will spend exactly n days there. Now you have to choose which attractions you would like to visit on each of these days.

On the day i there are k_i attractions, the j-th of them will bring a_{ij} pleasure to you and b_{ij} pleasure to your friend. Note, that you will together visit exactly one attraction per day.

You would like to get maximum possible pleasure from the journey, so you need to choose right attractions to visit. If the total pleasure you will get from all visited attractions is A, and the total pleasure your friend will get is B, then the pleasure of the journey is $\min(A, B)$.

Compute the maximal possible pleasure of the journey if you make optimal choice of attractions to visit.

Input

The first line of input contains one integer n ($2 \le n \le 10^4$) — the number of days. The following lines contain descriptions of attractions for each of these n days.

The first line of the description of attractions on the i-th day contains an integer k_i ($1 \le k_i \le 3 \cdot 10^4$) — the number of events on the i-th day.

Each of the following k_i lines contains two numbers a_{ij} and b_{ij} ($0 \le a_{ij}, b_{ij} \le 50$) — the pleasure of the attraction for you and your friend, respectively.

Let us denote $K = k_1 + k_2 + \ldots + k_n$. It is guaranteed that $K \le 3 \cdot 10^4$.

Output

Output a single number — the answer to the task.

Example

standard input	standard output
3 2 0 50 50 12 3 33 0 10 39 0 31 2 0 29 11 10	61

Scoring

Group	Points	Add. constraints		Required groups	Comment
		n	K		
0	0	–	–	–	Tests from the Statement.
1	23	$n \le 20$	$K \le 40$	0	–
2	15	$n \le 1000$	$K \le 2000$	0, 1	–
3	12	–	–	0	$a_{ij}, b_{ij} \le 10$
4	11	–	$K \le 10^4$	0	–
5	39	–	–	$1 - 4$	–