1. k-th divisor

You are given two integers n and k. Find k-th smallest divisor of n, or report that it doesn't exist.

Divisor of n is any such natural number, that n can be divided by it without remainder.

Input

The first line contains two integers n and k (1 <= n <= 10, 1 <= k <= 10)

Output

If n has less than k divisors, output -1.

Otherwise, output the k-th smallest divisor of n.

Examples

Input 1

4 2

Output 1

2

Input 2

5 3

Output 2

-1

Input 3

12 5

Output 3

6

Note

In the first example, number 4 has three divisors: 1, 2 and 4. The second one is 2.

In the second example, number 5 has only two divisors: 1 and 5. The third divisor doesn't exist, so the answer is -1.

Testcase:

1000000000000000 100

2. The football fest

The very famous football club Manchester United decided to popularize football in India by organizing a football fest. The fest had many events for different sections of people. For the awesome coders , there was an event called PASS and BACK. In this event, the coders were given N passes and players having ids between 1 and 1000000.

Initially some player with a given id had the ball in his possession. The coders had to make a program to display the id of the player who possessed the ball after exactly N passes.

Description of the passes:

There were two kinds of passes:

1. P ID
2. B

Explanation :

for the first kind of pass, the player in possession of the ball passes the ball to player with id=ID while for the second kind of a pass, the player in possession of the ball passes the ball back to the player who had passed the ball to him.

NOTE:

It is guaranteed that the given order of all the passes will be a valid order.

INPUT :

The first line of the input contains the number of test cases. For each test case, two space separated integers N and ID (of the player possessing the ball in the very beginning). N lines follow describing the passes. (for description of the passes, refer the statement above.)

OUTPUT :

Output to each test case should be a single line containing the 'player' ID, of the player who possesses the ball after N passes.

CONSTRAINTS :

1≤T≤100

1≤N≤100000

1≤ID≤1000000

Input :
1
7 23
P 86
P 63
P 23
P 99
P 9
B
B
Output :
Player 9

Testcase:
3
10 867723
P 799770
B
B
P 634289

B
P 321133
P 45496
B
P 830224
B
8 74784
P 719163
P 237316
P 347464
B
P 790535
P 903418
P 630191
B
8 801323
P 828019
P 764587
P 625357
B
P 739801
P 898914
P 801476
P 726939
8 936070
P 341524
P 812150
B
P 552768
P 468984
P 767837
B
P 605947

3. Soda Bottles

Aneesh has n bottles of soda left after his birthday. Each bottle is described by two values: remaining amount of soda a_i and bottle volume b_i ($a_i \leq b_i$).

Aneesh has decided to pour all remaining soda into minimal number of bottles, moreover he has to do it as soon as possible. Aneesh spends x seconds to pour x units of soda from one bottle to another.

Aneesh too tired after his birthday bash asks you to help him to determine k — the minimal number of bottles to store all remaining soda and t — the minimal time to pour soda into k bottles. A bottle can't store more soda than its volume. All remaining soda should be saved.

Input

The first line contains positive integer n ($1 \leq n \leq 100$) — the number of bottles.

The second line contains n positive integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 100$), where a_i is the amount of soda remaining in the i-th bottle.

The third line contains n positive integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_i \leq 100$), where b_i is the volume of the i-th bottle.

It is guaranteed that $a_i \leq b_i$ for any i.

Output

The only line should contain two integers k and t, where k is the minimal number of bottles that can store all the soda and t is the minimal time to pour the soda into k bottles.

Examples

input

```
4
3 3 4 3
4 7 6 5
```

output

```
2 6
```

input

```
2
1 1
100 100
```

output

```
1 1
```

Test Case:

```
5
10 30 5 6 24
10 41 7 8 24
```

Note

In the first example Aneesh can pour soda from the first bottle to the second bottle. It will take 3 seconds. After it the second bottle will contain $3 + 3 = 6$ units of soda. Then he can pour soda from the fourth bottle to the second bottle and to the third bottle: one unit to the second and two units to the third. It will take $1 + 2 = 3$ seconds. So, all the soda will be in two bottles and he will spend $3 + 3 = 6$seconds to do it.

4. Ping Pong

Dimitrij Ovtcharov and Ma Long have played several table tennis sets.
Each set consists of several serves, each serve is won by one of the players, he receives one point and the loser receives nothing.
Once one of the players scores exactly k points, the score is reset and a new set begins.

Across all the sets Ma Long scored a points in total, and Dimitrij Ovtcharov scored b points.
Given this information, determine the maximum number of sets they could have played,
or that the situation is impossible.

Note that the game consisted of several complete sets.

Input
The first line contains three space-separated integers k, a and b
(1<=k<=10^9, 0<=a,b<=10^9, a+b>0).

Output
If the situation is impossible, print a single number -1.
Otherwise, print the maximum possible number of sets.

Sample case:

input
11 11 5
output
1

Note
Note that the rules of the game in this problem differ from the real table tennis game,
for example, the rule of "balance" (the winning player has to be at least two points ahead to win a set)
has no power within the present problem.

Final Test Case:

input
12938621 192872393 102739134

5. **Umbrella Distribution**

It is a really hot summer in Manipal and an NGO is planning on distributing umbrellas to the needy. Since one store cannot supply all the umbrellas they need, they approach different stores and buy a different number of umbrellas from them. The number of stores umbrellas were bought from is **N** and the number of teams to give out the umbrellas is also **N**.

The founder of the NGO believes in equality i.e. all teams should get **equal** number of umbrellas to distribute. To achieve this, he wanted to rearrange the umbrellas bought. All umbrellas from a store are in one bag.

But since it is getting late, you are asked to do this in minimum number of umbrellas moved.

Input Format

- The first line contains a single integer **N** denoting the number of stores visited. The second line contains **N** space-separated integers $\mathbf{U_1}$, $\mathbf{U_2}$,..., $\mathbf{U_N}$ denoting the number of umbrellas in the bag from the i^{th} store.

Output Format

- Output a single line containing number of minimum moves required such that each team gets equal number of umbrellas. If it is not possible then print -1.

Constraints

- **1** <= **N** <= **10^5**
- **1** <=$\mathbf{U_i}$ <=**10^6**

Example

Input 1:
5
5 2 7 4 5

Output 1:
-1

Input 2:
6
6 3 5 2 8 6

Output 2:
5

Explanation

Example case 2.

1. You move one umbrella from 1st bag to 2nd bag. (Distribution looks like 5 4 5 2 8 6).
2. You move one umbrella from 6th bag to 2nd bag. (Distribution looks like 5 5 5 2 8 5).
3. You move three umbrellas from 5th bag to 4th bag. (Distribution looks like 5 5 5 5 5 5).

Final distribution looks like 5 5 5 5 5 5 which took place in 5 moves.

Test case

40

86 8698 258 384 3983 1622 2844 9877 972 38539 29482 30090 112 34575 35735 9744 47644 76543 35964 78653 88764 33576 443578 987 3345 65437 8975 2245 8426 8864 7654 8764 6549 8875 34577 876 7654 8763 33456 87630

6. Measurement Error

Your professor has made **V** measurements to calculate the name of your soulmate. You are required to find the average of those values. After all your effort, the professor tells you he just found out that due to instrument error (and your professor's old age), the smallest and the largest **A** values are incorrect and need to be excluded from the observation.

It seems like too much effort now but you remember how your friends tag you in 'Forever Alone' posts and want to prove them wrong. Figure out how to do so before your professor leaves for DeeTee and forgets the algorithm to find your soulmate's name.

Input Format

- The first line contains two integers **V** and **A** denoting the number of measurements and the number of erroneous values. The second line contains **V** space-separated integers M_1, M_2,..., M_N denoting each measurement.

Output Format

- Output a single line containing the average of the measurements, with an error of less than 10^{-6}.

Constraints

- **1** <= **V** <= **10^4**
- **0** <=**2A** < **V**
- **-10^6** <= M_i <=**10^6**

Example

Input 1:
5 1
5 2 7 4 5

Output 1:
4.666667

Input 2:

5 2
2 9 -10 25 1

Output 2:
2.000000

Explanation

Example case 2.

Removing the two largest and two smallest elements just leaves 2, whose average is 2.

Test case

40 15

86 8698 258 384 3983 1622 2844 9877 972 38539 29482 30090 112 34575 35735 9744 47644 76543 35964 78653 88764 33576 443578 987 3345 65437 8975 2245 8426 8864 7654 8764 6549 8875 34577 876 7654 8763 33456 87630

7. Stripe

Once Bob took a paper stripe of n squares (the height of the stripe is 1 square). In each square he wrote an integer number, possibly negative. He became interested in how many ways exist to cut this stripe into two pieces so that the sum of numbers from one piece is equal to the sum of numbers from the other piece, and each piece contains positive integer amount of squares. Would you help Bob solve this problem?

Input

The first input line contains integer n (1<= n<= 10^5) — amount of squares in the stripe. The second line contains n space-separated numbers — they are the numbers written in the squares of the stripe. These numbers are integer and do not exceed 10000 in absolute value.

Output

Output the amount of ways to cut the stripe into two non-empty pieces so that the sum of numbers from one piece is equal to the sum of numbers from the other piece. Don't forget that it's allowed to cut the stripe along the square's borders only.

Examples

Input 1

3

1 1 1

Output 1

0

Input 2

2

0 0

Output 2

1

Testcase

9

1 5 -6 7 9 -16 0 -2 2

8.Save thy World!

Hackerland has n houses in the xy plane . You are given x and y coordinates of all the n houses. You plan to drop a bomb on hackerland. The bomb you possess has a destruction radius of D . Formally, if you drop the bomb at point (a,b), all houses within a distance D from this point will be destroyed. You decide to drop the bomb at a point selected uniformly randomly inside a circle of radius R, centered at (0,0). Find the expected value of the number of houses destroyed by the bomb.

Input
The first line contains n, D, and R respectively.
It is followed by n lines.The ith line contains two integers xi, and yi, representing x and y coordinates of the ith house.
Output

Output a single line containing the expected value of number of houses destroyed by the bomb to exactly 4 digits after the decimal point.
Constraints
n,D,R,xi,yi are integers

1<=n,D,R<=10^6
−10^6<=xi,yi<=10^6

Sample input:
1 2 1
0 0

Sample output:
1.0000

Because D=2, and R=1, no matter where you drop the bomb inside a circle of radius R centered at (0,0), the point (0,0) will always be destroyed. So one house is destroyed no matter where the bomb is dropped. Hence, expected value of number of houses destroyed is 1.

Test Case:

7 3 4
0 14
1 3
13 11
13 13
12 7
8 9
16 13

9. Delete! Delete! Delete!

Jeff Hardy is thinking of a joining the WWE(World Wrestling Environment) College of Wrestling where subjects are given numbers in the range 1 to M . But he finds that R wrestlers enrolled before him via the special NXT program and there are a few seats left now.

But Jeff's lucky that his brother Broken Matt has some contacts with the boss of the school Triple H. Unfortunately, Matt can get him a seat but for the subject with minimal value of Z . Where Z= m*N(m is the number of students enrolled in ith subject and N is the Fighting Points of the course found by adding the Fighting Points of the last 2 wrestlers enrolled)

The Game(Triple H) is very lazy and thus he hands him the subject list to find subject i (1<= i <= M) for which Z is minimal.

If a single wrestler has enrolled for the subject, the Fighting Points will be given by the Fighting Points of the lone wrestler. If there are no wrestlers , the Fighting Points for the subject is 0.If the value of Z for two subjects are same he is allowed to pick the one with lower subject number.

However there's a little twist. Jeff has a few buddies (W-1) from his school TNA (Total Nonsense Action) who wants to join WWE with him. Thus you have to help each of them pick a subject one after the other , beginning with Jeff. (Assuming all of his friends are given the same condition for seat selection as Jeff)

Oh the Broken Brilliance!

Time to Play the Game!

Input Format

The first line contains the numbers M, W and R where M denotes the number of subjects in that college, W represents Jeff and his buddies and R denotes the number of students who have already applied for the courses.

The second line consists of R integers F[i] which represents fighting ability of ith wrestler. Here, the ith wrestler chooses the ith course.

The third line consists of W integers K[i] which represents the fighting ability of Jeff and his friends.

Constraints

Constraints : 1 <= M <= 100000

1 <= W <= 100000

1 <= R <= M

1 <= K[i],F[i] <= 100000

Output Format

Print W space separated integers in a line to represent the subject number which Jeff and his buddies get to apply for.

Sample Input

5 4 4

2 8 5 1

9 10 5 1

Sample Output

5 4 1 3

Explanation

Nobody enrolled for the last subject. So the sum is 0 initially. Hence, Jeff will go for that subject. Now Jeff's first buddy will enrol for the 4th subject as its value of Z = 1, which is minimum of all Similary, second friend for 1st subject and finally 3rd for the 3rd subject as its value of Z = 5, which is minimum of all

Test Case:

10 10 10
2 8 5 1 10 5 9 9 3 5
2 5 1 8 6 5 1 10 1 6

10. Robbery

It is nighttime and Joe the Elusive got into the country's main bank's safe. The safe has n cells positioned in a row, each of them contains some amount of diamonds. Let's make the problem more comfortable to work with and mark the cells with positive numbers from 1 to n from the left to the right.

Unfortunately, Joe didn't switch the last security system off. On the plus side, he knows the way it works.

Every minute the security system calculates the total amount of diamonds for each two adjacent cells (for the cells between whose numbers difference equals 1). As a result of this check we get an n - 1 sums. If at least one of the sums differs from the corresponding sum received during the previous check, then the security system is triggered.

Joe can move the diamonds from one cell to another between the security system's checks. He manages to move them no more than m times between two checks. One of the three following operations is regarded as moving a diamond: moving a diamond from any cell to any other one, moving a diamond from any cell to Joe's pocket, moving a diamond from Joe's pocket to any cell. Initially Joe's pocket is empty, and it can carry an unlimited amount of diamonds. It is considered that before all Joe's actions the system performs at least one check.

In the morning the bank employees will come, which is why Joe has to leave the bank before that moment. Joe has only k minutes left before morning, and on each of these k minutes he can perform no more than m operations. All that remains in Joe's pocket, is considered his loot.

Calculate the largest amount of diamonds Joe can carry with him. Don't forget that the security system shouldn't be triggered (even after Joe leaves the bank) and Joe should leave before morning.

Input

The first line contains integers n, m and k (1 <= n <= 10^4, 1 <= m, k <= 10^9). The next line contains n numbers. The i-th number is equal to the amount of diamonds in the i-th cell — it is an integer from 0 to 10^5.

Output

Print a single number — the maximum number of diamonds Joe can steal.

Examples

Input 1

2 3 1
2 3

Output 1

0

Input 2

3 2 2
4 1 3

Output 2

2

Note

In the second sample Joe can act like this:

The diamonds' initial positions are 4 1 3.

During the first period of time Joe moves a diamond from the 1-th cell to the 2-th one and a diamond from the 3-th cell to his pocket.

By the end of the first period the diamonds' positions are 3 2 2. The check finds no difference and the security system doesn't go off.

During the second period Joe moves a diamond from the 3-rd cell to the 2-nd one and puts a diamond from the 1-st cell to his pocket.

By the end of the second period the diamonds' positions are 2 3 1. The check finds no difference again and the security system doesn't go off.

Now Joe leaves with 2 diamonds in his pocket.

Testcase

13 94348844 381845400

515 688 5464 155 441 9217 114 21254 55 9449 1800 834 384

11. Jumping Jack

Jack is working on his jumping skills recently. Currently he's located at point zero of the number line. He would like to get to the point x. In order to train, he has decided that he'll first jump by only one unit, and each subsequent jump will be exactly one longer than the previous one. He can go either left or right with each jump. He wonders how many jumps he needs to reach x.

Input

The input data consists of only one integer x (-10^9 <= x <= 10^9).

Output

Output the minimal number of jumps that Jack requires to reach x.

Examples

Input 1

2

Output 1

3

Input 2

6

Output 2

3

Testcase

-1000000000

12. Amazing Coders Meet

You probably know those quizzes in Sunday magazines: given the sequence 1, 2, 3, 4, 5, what is the next number? Sometimes it is very easy to answer, sometimes it could be pretty hard. Because these "sequence problems" are very popular, ACM wants to implement them into the "Free Time" section of their new WAP portal.

ACM programmers have noticed that some of the quizzes can be solved by describing the sequence by polynomials. For example, the sequence 1, 2, 3, 4, 5 can be easily understood as a trivial polynomial. The next number is 6. But even more complex sequences, like 1, 2, 4, 7, 11, can be described by a polynomial. In this case, 1/2.n^2-1/2.n+1 can be used. Note that even if the members of the sequence are integers, polynomial coefficients may be any real numbers.

Polynomial is an expression in the following form:

P(n) = aD.n^D+aD-1.n^D-1+...+a1.n+a0

If aD <> 0, the number D is called a degree of the polynomial. The zero function P(n) = 0 is usually defined to have degree -1.

Input

There is a single positive integer T on the first line of input (equal to about 5000). It stands for the number of test cases to follow. Each test case consists of two lines. First line of each test case contains two integer numbers S and C separated by a single space, 1 <= S < 100, 1 <= C < 100, (S+C) <= 100. The first number, S, stands for the length of the given sequence, the second number, C is the amount of numbers you are to find to complete the sequence.

The second line of each test case contains S integer numbers X1, X2, ... XS separated by a space. These numbers form the given sequence. The sequence can always be described by a polynomial P(n) such that for every i, Xi = P(i). Among these polynomials, we can find the polynomial Pmin with the lowest possible degree. This polynomial should be used for completing the sequence.

Output:

For every test case, your program must print a single line containing C integer numbers, separated by a space. These numbers are the values completing the sequence according to the polynomial of the lowest possible degree. In other words, you are to print values Pmin(S+1), Pmin(S+2), ... Pmin(S+C).

It is guaranteed that the results Pmin(S+i) will be non-negative and will fit into the standard integer type.

Input Case 1:

4
6 3
1 2 3 4 5 6
8 2
1 2 4 7 11 16 22 29
10 2
1 1 1 1 1 1 1 1 1 2
1 10
3

Output Case 1:

7 8 9
37 46
11 56
3 3 3 3 3 3 3 3 3 3

Test Case:
1
6 2
7 11 23 37 30 32

13. Complement

Understanding 2*'s complement* representation is fundamental to learning about Computer Science. Let's say you wrote down the 2's complement for each 32-bit integer in the inclusive range from A to B; how many 's would you write down in all?

Input Format

The first line contains T , the number of test cases.
The T subsequent lines each contain two space-separated integers, A and B, respectively.

Constraints

- T<=1000
- -2^31 <= A<=B<= 2^31-1

Output Format

On a new line for each of the T test cases, print the number of 1's in the 32-bit 's 2's complement representation for integers in the inclusive range from A to B .

Sample Input

```
3
-2 0
-3 4
-1 4
```

Sample Output

```
63
99
37
```

Explanation

Test Case 0:
-2 contains 31 ones followed by a zero.
-1 contains 32 ones.
0 contains 0 ones.
31+32+0=63, so we print 63 on a new line.
Test Case 1:
31+31+32+0+1+1+2+1=99, so we print 99 on a new line.
Test Case 2:
32+0+1+1+2+1=37, so we print 37 on a new line

Test case

Input

1

-7 7

14. <u>Sublime Strings</u>

You just started studying FLAT and are given a **uniformly randomly generated** string **S**, consisting of letters from the set **{"A", "B"}**. To get an A+ you need to check is a given string is sublime. A sublime string **S** has a string **T** that appears in **S** as a subsequence **exactly twice**.

In other words, you need to find such a string **T**, that there exist **exactly two** sets of indexes $i_1, i_2, ..., i_{|T|}$ and $j_1, j_2, ..., j_{|T|}$ such that there exists some **k**, where $i_k \neq j_k$ and $S_{\{i1...i|T|\}} = S_{\{j1...j|T|\}} = T$.

Input Format

- The only line contains a string **S**.

Output Format

- Output a single line containing the string **T** which makes **S** sublime or -1 if no such string exists.

Example

Input 1:
AAAA

Output 1:
-1

Input 2:
BAB

Output 2:
B

Explanation

Example case 2.

Two occurrences of "B" in "BAB" are {1} and {3} (1-based indexing).

Test case

AAABBABABABABBBBBBAAABBBABABABAB

15. WORLD MAP

We consider a geographical map with N countries numbered from 1 to N ($0 < N < 99$). For every country we know the numbers of other countries which are connected with its border. From every country we can reach to any other one, eventually crossing some borders. Write a program which determines whether it is possible to color the map only in two colors — red and blue in such a way that if two countries are connected their colors are different. The color of the first country is red. Your program must output one possible coloring for the other countries, or show, that such coloring is impossible.

Input

On the first line is written the number N. On the following N lines, the i-th line contains the countries to which the i-th country is connected. Every integer on this line is bigger than i, except the last one which is 0 and marks that no more countries are listed for country i. If a line contains 0, that means that the i-th country is not connected to any other country, which number is larger than i.

Output

The output contains exactly one line. If the coloring is possible, this line must contain a list of zeros and ones, without any separators between them. The i-th digit in this sequence is the color of the i-th country. 0 corresponds to red color, and one — to blue color. If a coloring is not possible, output the integer −1.

Input

3

2 0

3 0

0

Output

010

Testcase:

6

2 3 0

4 0

5 0

6 0

6 0

0