Codeforces Round #273 (Div. 2)

A. Initial Bet

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

There are five people playing a game called "Generosity". Each person gives some non-zero number of coins b as an initial bet. After all players make their bets of b coins, the following operation is repeated for several times: a coin is passed from one player to some other player.

Your task is to write a program that can, given the number of coins each player has at the end of the game, determine the size b of the initial bet or find out that such outcome of the game cannot be obtained for any positive number of coins b in the initial bet.

Input

The input consists of a single line containing five integers c_1, c_2, c_3, c_4 and c_5 — the number of coins that the first, second, third, fourth and fifth players respectively have at the end of the game ($0 \le c_1, c_2, c_3, c_4, c_5 \le 100$).

Output

Print the only line containing a single positive integer b — the number of coins in the initial bet of each player. If there is no such value of b, then print the only value "-1" (quotes for clarity).

Sample test(s)

input
2 5 4 0 4
output
3

input
4 5 9 2 1
output
-1

Note

In the first sample the following sequence of operations is possible:

1. One coin is passed from the fourth player to the second player;
2. One coin is passed from the fourth player to the fifth player;
3. One coin is passed from the first player to the third player;
4. One coin is passed from the fourth player to the second player.

B. Random Teams

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

n participants of the competition were split into m teams in some manner so that each team has at least one participant. After the competition each pair of participants from the same team became friends.

Your task is to write a program that will find the minimum and the maximum number of pairs of friends that could have formed by the end of the competition.

Input

The only line of input contains two integers n and m, separated by a single space ($1 \le m \le n \le 10^9$) — the number of participants and the number of teams respectively.

Output

The only line of the output should contain two integers k_{min} and k_{max} — the minimum possible number of pairs of friends and the maximum possible number of pairs of friends respectively.

Sample test(s)

input
5 1
output
10 10

input
3 2
output
1 1

input
6 3
output
3 6

Note

In the first sample all the participants get into one team, so there will be exactly ten pairs of friends.

In the second sample at any possible arrangement one team will always have two participants and the other team will always have one participant. Thus, the number of pairs of friends will always be equal to one.

In the third sample minimum number of newly formed friendships can be achieved if participants were split on teams consisting of 2 people, maximum number can be achieved if participants were split on teams of 1, 1 and 4 people.

C. Table Decorations

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

You have r red, g green and b blue balloons. To decorate a single table for the banquet you need exactly three balloons. Three balloons attached to some table shouldn't have the same color. What maximum number t of tables can be decorated if we know number of balloons of each color?

Your task is to write a program that for given values r, g and b will find the maximum number t of tables, that can be decorated in the required manner.

Input

The single line contains three integers r, g and b ($0 \le r, g, b \le 2 \cdot 10^9$) — the number of red, green and blue baloons respectively. The numbers are separated by exactly one space.

Output

Print a single integer t — the maximum number of tables that can be decorated in the required manner.

Sample test(s)

input
5 4 3
output
4

input
1 1 1
output
1

input
2 3 3
output
2

Note

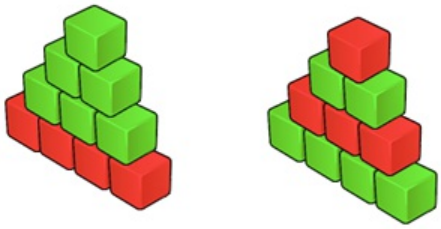
In the first sample you can decorate the tables with the following balloon sets: "`rgg`", "`gbb`", "`brr`", "`rrg`", where "`r`", "`g`" and "`b`" represent the red, green and blue balls, respectively.

D. Red-Green Towers

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

There are r red and g green blocks for construction of the *red-green* tower. *Red-green* tower can be built following next rules:

- Red-green tower is consisting of some number of levels;
- Let the red-green tower consist of n levels, then the first level of this tower should consist of n blocks, second level — of n - 1 blocks, the third one — of n - 2 blocks, and so on — the last level of such tower should consist of the one block. In other words, each successive level should contain one block less than the previous one;
- Each level of the red-green tower should contain blocks of the same color.



Let h be the maximum possible number of levels of red-green tower, that can be built out of r red and g green blocks meeting the rules above. The task is to determine how many different red-green towers having h levels can be built out of the available blocks.

Two red-green towers are considered different if there exists some level, that consists of red blocks in the one tower and consists of green blocks in the other tower.

You are to write a program that will find the number of different red-green towers of height h modulo $10^9 + 7$.

Input

The only line of input contains two integers r and g, separated by a single space — the number of available red and green blocks respectively $(0 \le r, g \le 2 \cdot 10^5, r + g \ge 1)$.

Output

Output the only integer — the number of different possible red-green towers of height h modulo $10^9 + 7$.

Sample test(s)

input
4 6
output
2

input
9 7
output
6

input
1 1
output
2

Note

The image in the problem statement shows all possible red-green towers for the first sample.

E. Wavy numbers

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A *wavy number* is such positive integer that for any digit of its decimal representation except for the first one and the last one following condition holds: the digit is either strictly larger than both its adjacent digits or strictly less than both its adjacent digits. For example, numbers 35270, 102, 747, 20 and 3 are *wavy* and numbers 123, 1000 and 2212 are not.

The task is to find the k-th **smallest** *wavy number* r that is divisible by n for the given integer values n and k.

You are to write a program that will find the value of r if it doesn't exceed 10^{14}.

Input

The only line of input contains two integers n and k, separated by a single space ($1 \le n, k \le 10^{14}$).

Output

Your task is to output the only integer r — the answer to the given problem. If such number does not exist or it is larger than 10^{14}, then print "-1" (minus one without the quotes) instead.

Sample test(s)

input
123 4
output
1845

input
100 1
output
-1

input
97461 457
output
1805270103

Note

The values of the first four *wavy numbers* that are divisible by n for the first sample are: 492, 615, 738 и 1845.