

Hong Kong Girls' Olympiad in Informatics 2025/26

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
G261	Sushi Go	1.000 s	256 MB	9 + 9 + 15 + 16 + 18 + 25 + 8
G262	Removing Letters	1.000 s	256 MB	11 + 22 + 14 + 21 + 32
G263	Screws	1.000 s	256 MB	7 + 9 + 17 + 24 + 23 + 20
G264	Infinite Whack-A-Mole	1.000 s	256 MB	13 + 22 + 18 + 13 + 16 + 18
G265	Mini Metro	1.000 s	256 MB	7 + 10 + 17 + 22 + 23 + 21
G266	Full Speed Ahead	1.000 s	256 MB	8 + 14 + 20 + 19 + 21 + 18

Notice:

Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

For some problems 64-bit integers may be required. In C++ it is `long long` and its token for `scanf`/`printf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

Organiser



Co-organiser



Sponsor



G261 - SUSHI GO

Time Limit: 1.000 s / Memory Limit: 256 MB

Alice and Bob are eating in a sushi shop. In the sushi shop, there are different types of sushi, as well as some side dishes. The prices of different sushi are categorized by their dish color, where white is the cheapest and black is the most expensive. The menu is as follows:

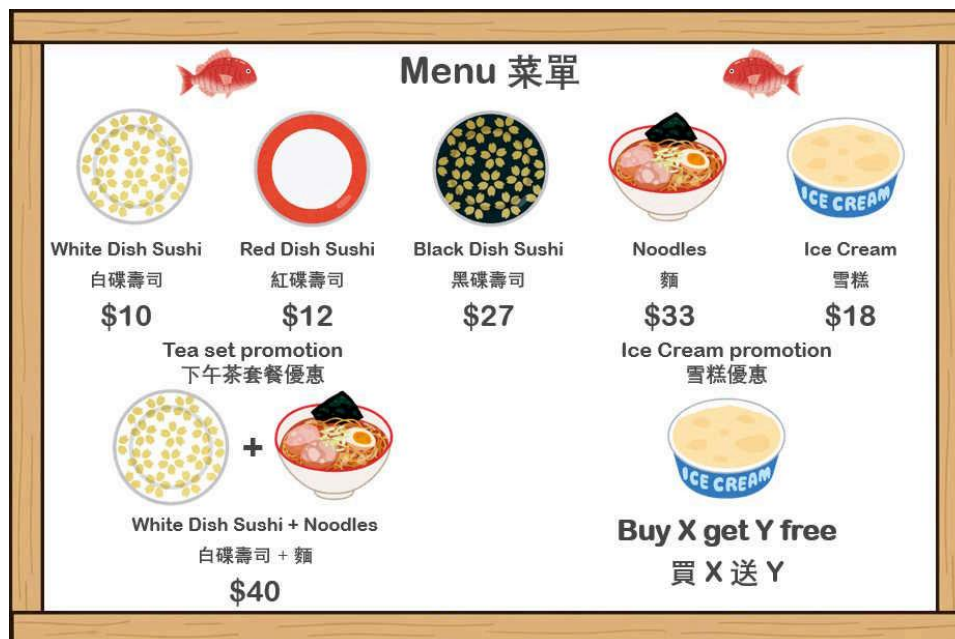


Figure 1: The menu and promotions of the sushi shop.

Formally, the sushi shop offers the following items:

Index	Name	Price
1	White Dish Sushi	\$10
2	Red Dish Sushi	\$12
3	Black Dish Sushi	\$27
4	Noodles	\$33
5	Ice Cream	\$18

Additionally, the sushi shop is offering some special promotions:

- From 14:00 to 16:59 (inclusive), there is a tea set sold at price \$40, which includes one White Dish Sushi and one bowl of Noodles. This tea set is only available during this time.
- There is a buy X get Y free promotion for Ice Cream. If $Y = 0$, this special promotion is not active.

Each promotion can be used multiple times. For example, if Alice and Bob purchase $2 \times X$ cups of ice cream, they will get $2 \times Y$ cups of ice cream for free.

Now, Alice and Bob arrived at the shop at time $H:M$. They want to order their favourite dishes. More specifically, they want **at least** A_1 dishes of the item with index 1, at least A_2 dishes of the item with index 2, and so on, up to at least A_5 dishes of the item with index 5. Your task is to calculate the minimum cost to purchase all items they want, given that they order once immediately at the time they arrived.

INPUT

The first 5 lines of input each contains one integer. The i -th of these lines ($1 \leq i \leq 5$) contains the integer A_i , representing the minimum number of dishes of the item with index i that Alice and Bob want.

The next line of input contains two integers H and M , representing the time they arrived at the shop in 24-hour format without leading zeros.

The next line of input contains two integers X and Y , representing the details of the ice cream promotion.

OUTPUT

Output a single integer, the minimum cost to purchase all the items they want immediately.

Hints

No need to validate inputs. Do not output prompts. (e.g.: `Please enter X and Y:`, `The answer is:`) Do not output any units.

You may test your program with the following samples. Your output should be exactly identical to the sample output.

Tasks are divided into subtasks. Partially correct solutions will still get some marks.

SAMPLE TESTS

	Input	Output
1	<pre> 1 4 0 1 1 12 0 1 0 </pre>	109

Alice and Bob arrived at the shop at `12:00`, and $Y = 0$, so both special promotions are not active.

The minimum total cost is $\$10 + 4 \times \$12 + \$0 + \$33 + \$18 = \109 .

This sample satisfies the constraints of subtask 3, 4, 6 and 7.



Figure 2: Illustration of Sample 1.

	Input	Output
2	4 0 0 2 0 15 12 1 0	100

Only the tea set promotion is active.

They should purchase 2 sets of tea set for \$80, and 2 white dish sushi separately for \$20. The total cost is \$100.

This sample satisfies the constraints of subtask 4 and 7.



Figure 3: Illustration of Sample 2.

3	0 0 0 0 7 12 0 1 1	72
---	---	----

Only the buy 1 get 1 free ice cream promotion is active.

They can purchase 4 ice cream, and get 4 ice cream for free. The total cost is $4 \times \$18 = \72 .

This sample satisfies the constraints of subtask 5, 6 and 7.



Figure 4: Illustration of sample 3.

4	1 0 0 1 6 12 0 3 2	115
---	---	-----

Only the buy 3 get 2 free ice cream promotion is active.

They can purchase 4 ice cream, and get 2 ice cream for free, and also purchase a white dish sushi and a bowl of noodles separately. The total cost is $\$10 + \$33 + 4 \times \$18 = \115 .

This sample satisfies the constraints of subtask 6 and 7.

	Input	Output
5	7 20 4 5 40 16 48 4 2	1072

Both the tea set promotion and the buy 4 get 2 free ice cream promotion are active.
This sample satisfies the constraints of subtask 7.

SUBTASKS

For all cases:

$$0 \leq A_i \leq 100 \text{ for all } 1 \leq i \leq 5$$

$$10 \leq H \leq 21$$

$$0 \leq M \leq 59$$

$$1 \leq X \leq 100$$

$$0 \leq Y \leq 100$$

	Points	Constraints
1	9	$H:M = \boxed{12:00}$ $Y = 0$ $A_1 = 0, A_2 = 2, A_3 = 0, A_4 = 1, A_5 = 1$
2	9	$H:M = \boxed{15:00}$ $X = Y = 1$ $A_1 = 2, A_2 = 1, A_3 = 1, A_4 = 1, A_5 = 3$
3	15	$H:M = \boxed{12:00}$ $Y = 0$
4	16	$Y = 0$
5	18	$H:M = \boxed{12:00}$ $X = Y = 1$
6	25	$H:M = \boxed{12:00}$
7	8	No additional constraints

G262 - REMOVING LETTERS

Time Limit: 1.000 s / Memory Limit: 256 MB

Bob loves groups of three strings which are **beautiful**! A group of three strings is said to be beautiful if and only if, after sorting the characters within each string alphabetically, the three resulting strings are equal.

- The group $\boxed{\text{AABC}}, \boxed{\text{ACBA}}, \boxed{\text{BAAC}}$ is beautiful, as they all become $\boxed{\text{AABC}}$ after sorting.
- The group $\boxed{\text{AABC}}, \boxed{\text{ABCA}}, \boxed{\text{BA}}$ is not beautiful, as they become $\boxed{\text{AABC}}, \boxed{\text{AABC}}$, and $\boxed{\text{AB}}$ respectively after sorting, which are not equal.
- Note that a group of three empty strings is considered beautiful.

As Bob's friend, you plan to gift Bob a group of beautiful strings for his birthday.

Currently, you have a group of three strings of uppercase letters, namely S_1 , S_2 , and S_3 , which may or may not be beautiful. To make sure Bob loves your gift, you can perform the following operation to make S_1 , S_2 , and S_3 a group of beautiful strings:

- Choose any uppercase letter X .
- Choose two out of the three strings (S_1 and S_2 , S_1 and S_3 , or S_2 and S_3).
- Remove all occurrences of X from the two chosen strings. Note that the other string is not modified.

For example, if $S_1 = \boxed{\text{ABBGCD}}$, $S_2 = \boxed{\text{AABBCD}}$, and $S_3 = \boxed{\text{DCA}}$, and you choose to remove $\boxed{\text{A}}$ from S_1 and S_2 . Then S_1 and S_2 become $\boxed{\text{BBGCD}}$ and $\boxed{\text{BBCD}}$ respectively, and S_3 remains unchanged. If $S_1 = \boxed{\text{A}}$, $S_2 = \boxed{\text{B}}$, and $S_3 = \boxed{\text{C}}$, and you choose to remove $\boxed{\text{C}}$ from S_1 and S_3 . Then S_3 becomes an empty string, while S_1 and S_2 remain unchanged.

Please find the minimum number of operations needed to make S_1 , S_2 , and S_3 a beautiful group.

INPUT

The input consists of 3 lines, each line consists of a string of uppercase letters, representing S_1 , S_2 , and S_3 respectively.

OUTPUT

Output a single integer, the minimum number of operations needed to make S_1 , S_2 , and S_3 a beautiful group.

SAMPLE TESTS

	Input	Output
1	ABBGCD AABBCD DCA	4

To make the group beautiful, perform the following operations:

- Remove $\boxed{\text{A}}$ from S_1 and S_2 .
- Remove $\boxed{\text{A}}$ from S_1 and S_3 .
- Remove $\boxed{\text{B}}$ from S_1 and S_2 .
- Remove $\boxed{\text{G}}$ from S_1 and S_2 .

After sorting in alphabetical order, S_1 , S_2 , and S_3 are all equal to $\boxed{\text{CD}}$.

It can be proven that 4 is the minimum number of operations needed.

Input

Output

2

Z

Z

Z

0

The group is already beautiful at the beginning, no operation is needed.

3

Q

W

E

3

To make the group beautiful, perform the following operations:

- Remove

Q

 from S_1 and S_2 .
- Remove

W

 from S_2 and S_3 .
- Remove

E

 from S_1 and S_3 .

S_1 , S_2 , and S_3 all become empty strings.

It can be proven that 3 is the minimum number of operations needed.

SUBTASKS

For all cases:
 Let $|S|$ be the length of S
 $1 \leq |S_1|, |S_2|, |S_3| \leq 10^5$
 S_1, S_2 , and S_3 consists of uppercase letters only

	Points	Constraints
1	11	$S_2 = S_3$ $ S_1 = S_2 = S_3 = 1$
2	22	$S_2 = S_3$ $ S_1 = 2$ $ S_2 = S_3 = 1$
3	14	$S_2 = S_3$ Each letter appears in a string at most once
4	21	Each letter appears in a string at most once
5	32	No additional constraints

G263 - SCREWS

Time Limit: 1.000 s / Memory Limit: 256 MB

An unexpected power failure has occurred during HKGOI! Bob, a professional electrician, has been called in for an urgent fix.

After careful inspection, it is found that N screws, numbered from 1 to N , in the fuse box have become loose. To fix this, Bob must tighten them **in sequential order**, from screw 1 to screw N . The i -th screw has a diameter of D_i .

As a professional electrician, Bob has a collection of 10^9 screwdrivers, one for each integer size from 1 to 10^9 .

A screw can only be tightened if the screwdriver's size is compatible with the screw's diameter. Bob knows the tolerance threshold is K , meaning that a screw of diameter x can be tightened with a screwdriver of size y if and only if $x - K \leq y \leq x + K$.

Bob can only use one screwdriver at a time. Tightening a screw takes no time. He can, however, swap his current screwdriver for any other one from his collection anytime. Each swap takes **one minute** to find the new tool. Choosing the very first screwdriver takes no time.

Since the organizers of HKGOI want the competition to resume as soon as possible, help Bob find the minimum time (in minutes) required for him to tighten all the screws.

INPUT

The first line contains an integer N , the number of screws to be tightened.

The second line contains an integer K , the tolerance threshold.

The next N lines each contain an integer. The i -th of these lines contains an integer D_i , the diameter of the i -th screw.

OUTPUT

Output a single integer, the minimum time (in minutes) that is required for Bob to tighten all the screws.

SAMPLE TESTS

	Input	Output
1	7 2 2 3 1 5 6 100 3	3

There are $N = 7$ loose screws with diameters $[2, 3, 1, 5, 6, 100, 3]$ in order. The tolerance threshold K is equal to 2. Bob can tighten all screws in 3 minutes by swapping the screwdriver 3 times as follows:

- Use the screwdriver of size 1 to tighten screw 1 to 3.
- Swap the screwdriver of size 1 with the screwdriver of size 6 and use it to tighten screw 4 to 5.
- Swap the screwdriver of size 6 with the screwdriver of size 100 and use it to tighten screw 6.
- Swap the screwdriver of size 100 with the screwdriver of size 3 and use it to tighten screw 7.

Although there exist other ways to tighten the screws, it can be shown that 3 minutes is the minimum time required. This sample satisfies the constraint of subtask 6.

Input

Output

2

2	0
6	
1	
6	

Bob can use a screwdriver of size 6 to tighten all screws, so he does not need to perform any swaps.
 This sample satisfies the constraints of subtasks 1, 2, 4 and 6.

3

6	2
2	
1	
3	
4	
10	
15	
17	

This sample satisfies the constraints of subtasks 5 and 6.

4

4	2
0	
2	
1	
1	
2	

This sample satisfies the constraints of subtasks 3 and 6.

SUBTASKS

For all cases:

$$1 \leq N \leq 10^5$$

$$0 \leq K \leq 10^9$$

$$1 \leq D_i \leq 10^9 \text{ for all } 1 \leq i \leq N$$

Points

Constraints

1

7

$D_i \leq K$ for all $1 \leq i \leq N$

2

9

$N = 2$

3

17

$K = 0$

4

24

The answer must be either 0 or 1

5

23

$D_{i-1} \leq D_i$ for all $2 \leq i \leq N$

6

20

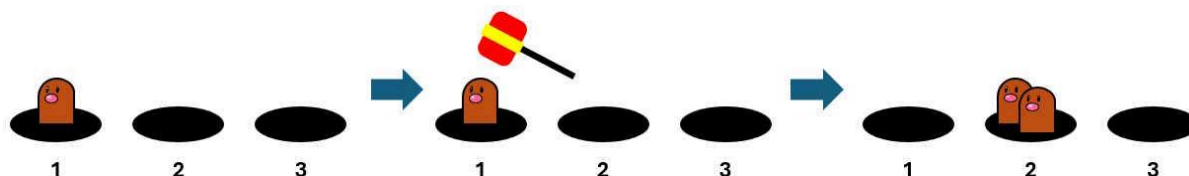
No additional constraints

G264 - INFINITE WHACK-A-MOLE

Time Limit: 1.000 s / Memory Limit: 256 MB

Alice is playing a new arcade game called "Infinite Whack-A-Mole".

In this game, moles can appear in $N + 1$ different holes labelled $1, 2, \dots, N + 1$ and initially, there is only one mole in hole 1. Whenever Alice whacks a mole in hole i where $1 \leq i \leq N$, it disappears, and two new moles appear in hole $i + 1$. You cannot whack moles in hole $N + 1$.



Alice whacks a mole in hole 1. After that, it disappears and two moles appear in hole 2.

Alice wins the game if she is able to get exactly M moles in hole N . Help her determine if that is possible, and if so, the minimum number of whacks needed.

INPUT

The first line of the input consists of an integer N , the hole where moles are counted.

The second line of the input consists of an integer M , the number of moles needed for Alice to win.

OUTPUT

If it is impossible to get exactly M moles in hole N , output -1 . Otherwise output the minimum number of whacks needed.

SAMPLE TESTS

	Input	Output
1	5 6	7

Alice can get exactly 6 moles in hole 5 by whacking moles in this order:

- Whack a mole in hole 1 using one whack. Two moles then appear in hole 2
- Whack a mole in hole 2 using one whack. Two moles then appear in hole 3.
- Whack two moles in hole 3 using two whacks. Four moles then appear in hole 4.
- Whack three moles in hole 4 using three whacks. Six moles then appear in hole 5, and Alice wins the game.

It can be proven that this is the least number of whacks needed for Alice to win. Multiple whacks are needed for the last two moves as we can only whack one mole each time.

	Input	Output			
2	<table><tr><td>3</td></tr><tr><td>3</td></tr></table>	3	3	<table><tr><td>4</td></tr></table>	4
3					
3					
4					

Alice can get exactly 3 moles in hole 3 by whacking moles in this order:

- Whack a mole in hole 1 using one whack. Two moles then appear in hole 2.
- Whack a mole in hole 2 using one whack. Two moles then appear in hole 3.
- Whack a mole in hole 3 using one whack. Two moles then appear in hole 4, and hole 3 has one mole left.
- Whack another mole in hole 2 using one whack. Two more moles then appear in hole 3 for a total of three moles, and Alice wins the game.

It can be proven that this is the least number of whacks needed for Alice to win.

3	<table><tr><td>8</td></tr><tr><td>5</td></tr></table>	8	5	11
8				
5				
4	<table><tr><td>3</td></tr><tr><td>10</td></tr></table>	3	10	-1
3				
10				
5	<table><tr><td>1</td></tr><tr><td>1</td></tr></table>	1	1	0
1				
1				

SUBTASKS

For all cases:
 $1 \leq N, M \leq 10^9$

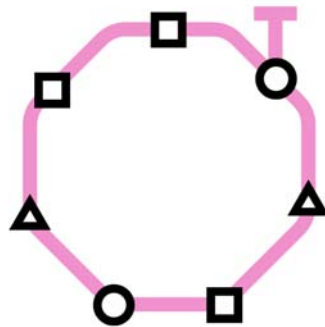
	Points	Constraints
1	13	$M \leq 2$
2	22	$N, M \leq 1000$
3	18	$M = 2^k$ for some integer k It is always possible for Alice to win
4	13	$M = 2^k$ for some integer k
5	16	$N \leq 10^5$
6	18	No additional constraints

G265 - MINI METRO

Time Limit: 1.000 s / Memory Limit: 256 MB

Bob is a huge fan of the game Mini Metro. Today, he wants to design the most efficient metro line possible.

After playing the game for hours, Bob has set up a circular metro line with N stations. The stations are numbered from 1 to N in clockwise order, where the metro stops at the N stations via the route: $1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow N \rightarrow 1 \rightarrow \dots$. The metro will operate along this circular route forever. It takes the metro 1 unit of time to travel from one station to another.



The passengers can be classified as K different types, numbered from 1 to K . Correspondingly, the stations can be classified as K different types as well. A passenger may start in any of the N stations and take the metro. The passenger of type i gets off the metro once it arrives at the first station classified as type i . If a passenger of type i starts at a station of type i , they do not take the metro at all.

Currently, Bob has to assign types to the N stations. He must assign A_1 stations as type 1, A_2 stations as type 2, $\dots$, and so on. It is guaranteed that the sum of A_i is exactly N .

Your goal is to help Bob optimally assign each station a type so that the maximum time a passenger would need to stay on the metro in the worst case is minimised, considering all possible starting stations and all passenger types.

INPUT

The first line of input consists of two integers, N and K .

The second line contains K integers $A_1, A_2, \dots, A_K$, the number of stations required for each type.

OUTPUT

On the first line, output an integer T , the maximum time that any passenger would need to stay on the metro in an optimal assignment of station types.

Then output one line containing N integers. The i -th integer, B_i , should be the type assigned to station i .

If there are multiple optimal assignments of station types, you may output any of them.

SAMPLE TESTS

	Input	Output				
1	<table><tr><td>7 3</td></tr><tr><td>2 2 3</td></tr></table>	7 3	2 2 3	<table><tr><td>3</td></tr><tr><td>1 2 3 3 1 2 3</td></tr></table>	3	1 2 3 3 1 2 3
7 3						
2 2 3						
3						
1 2 3 3 1 2 3						

- A type 1 passenger starting at station 2 will get off at station 5.
The route taken is $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, taking 3 units of time.
- A type 3 passenger starting at station 2 will get off at station 3.
The route taken is $2 \rightarrow 3$, taking 1 unit of time.
- A type 2 passenger starting at station 2 will not take the metro at all, taking 0 units of time.

It can be proven that no passenger stays on the metro for more than 3 units of time in the worst case, and that there does not exist an assignment of types where all passengers spend less than 3 units of time on the metro.

2	7 3	3
	2 2 3	2 3 1 3 2 1 3

This is another optimal assignment of station types for the above sample test.

3	6 2	5
	5 1	2 1 1 1 1 1

4	3 3	2
	1 1 1	3 1 2

SUBTASKS

For all cases:

$$2 \leq K \leq N \leq 100000$$

$$1 \leq A_i \leq N \text{ for all } 1 \leq i \leq K$$

$$A_1 + A_2 + \dots + A_K = N$$

	Points	Constraints
1	7	$K = 2$ $A_1 = A_2$
2	10	$A_1 = A_2 = \dots = A_K$
3	17	$K = 2$ $A_1 \leq A_2 \leq 2 \times A_1$
4	22	$K = 2$
5	23	$A_1 \leq A_2 \leq \dots \leq A_K \leq 2 \times A_1$
6	21	No additional constraints

G266 - FULL SPEED AHEAD

Time Limit: 1.000 s / Memory Limit: 256 MB

Alice lives in the world of HKGOI, which is represented by a one-dimensional number line. Alice's home is at point 0, and rumours say that there is a piece of gold located at point $D > 0$ of the number line.

Alice has a car to drive from point 0 to point D . The car has a maximum speed of M per hour, and can accelerate or decelerate by at most A per hour between any two consecutive hours.

Unfortunately, the car is rather old and has quite a few limitations. It cannot drive backwards, and moreover, it only operates at non-negative integer speeds. Also, due to a manufacturing error, when Alice starts the engine at point 0, the default speed is automatically set as S_0 per hour (which may or may not be 0).

Alice is so impatient to get that piece of gold, and so she wonders whether it is possible for her to drive to point D and completely stop her car there. If it is possible, she also wants to know the minimum time required to do so.

Formally, if we denote the speed of Alice's car during hour $i \geq 1$ to be S_i , then a T -hour-long drive from point 0 to point D should satisfy all of the following conditions:

- **speed limit:** the speed is always a non-negative integer no larger than M , so $0 \leq S_i \leq M$ for every $i \geq 1$;
- **limited acceleration:** the change in speed between consecutive hours must be no larger than A , meaning $-A \leq S_{i-1} - S_i \leq A$ for every $i \geq 1$ (in particular, when $i = 1$, the change from the initial speed S_0 must also be no larger than A);
- **complete stopping:** the car must completely stop after the drive, meaning the speed must be able to become 0 immediately after T hours, so $S_T \leq A$;
- **correct total distance:** the car must arrive exactly at point D , so $S_1 + S_2 + \dots + S_T = D$.

More rumours suggest that if you manage to help Alice determine the minimum time required to get from point 0 to point D , she might — just might — mould the piece of gold and cast it into a HKGOI medal for you as a gift. Hmm...

INPUT

The first line of the input contains a single positive integer D , denoting the distance to be driven by Alice.

The second line of the input contains a single non-negative integer S_0 , denoting the default speed of the car.

The third line of the input contains a single positive integer M , denoting the maximum speed of the car.

The fourth line of the input contains a single positive integer A , denoting the maximum acceleration and deceleration of the car.

OUTPUT

If it is impossible for Alice to drive from point 0 to point D satisfying all conditions, output $\boxed{-1}$.

Otherwise, if a valid drive exists, output a single positive integer, denoting the minimum time required, in hours, to arrive and completely stop her car at point D .

SAMPLE TESTS

	Input	Output
1	7 2 10 2	3

A possible way for Alice to arrive and completely stop at point 7 which uses 3 hours is as follows: move at speed 3 during the first hour, speed 2 during the second hour, and speed 2 during the third hour. Afterwards, Alice can completely stop her car.

It can be proven that Alice needs at least 3 hours to arrive and completely stop at point 7.

2	20 4 7 3	4
---	---	--------------

Alice can arrive at point 20 after 4 hours if she drives at speed 5 during the first hour, speed 7 during the second hour, speed 5 during the third hour, and speed 3 during the fourth hour.

3	14 6 10 1	-1
---	--	---------------

Alice can never arrive and completely stop her car at point 14.

SUBTASKS

For all cases:

$$1 \leq D \leq 10^{18}$$

$$0 \leq S_0 \leq 10^9$$

$$1 \leq A \leq 10^9$$

$$\max(S_0, A) \leq M \leq 10^9$$

	Points	Constraints
1	8	$M = A$
2	14	$S_0 = M$ $A = 1$ $D, M \leq 10^6$
3	20	$S_0 = 0$ $A = 1$ $D, M \leq 10^6$
4	19	$S_0 = 0$
5	21	$A = 1$
6	18	No additional constraints