

Perfect Array

This problem was worth 1000 points and was ranked easy.

The author of this problem is Aryan V S.

Note: GitHub does not support LaTeX in Markdown. If you want a more readable version of the problem, download the PDF file instead.

Statement

Aryan and Dhruv are taking part in an event organised for 1337 h4ck3r5. There are two teams that are to be formed during the event - the Blue team and the Red team. The two teams will take part in a game.

Initially, all of the $2n$ participants were divided into the two teams. All members of the Blue team (n of them to be precise) decided to play together from the beginning because they believe that there lies strength in unity. Dhruv is the leader of the Blue team. However, the Red teamers (n of them) decide to play individually because they believe that they are the world's best hackers and can take down the Blue team alone. Aryan is the leader of the Red team.

The Blue teamers need to defend their network from being attacked by the Red teamers. After a few initial practice rounds of the game, the Red teamers learnt that they must combine forces if they want to take down the Blue team and that they aren't as 1337 as they thought.

At the i^{th} second, the Blue team can create a "Blinky Box" and increase their total CyberShield hitpoints by k . This always happens before the Red teamers commence their attack. Their initial Blinky Box CyberShield hitpoints are H .

At the i^{th} second, one of the Red teamer groups *can* (it is not necessary) combine forces with a group of other Red teamers. Initially, a Red teamer's CyberNuke strength is a_j ($1 \leq j \leq n$). However, if they join a group of one or more Red teamers, their CyberNuke strength becomes $a_j \cdot \text{size}(\text{group})$. Once a Red teamer is part of a group, they play with that group until the end of the game. It is possible that some group can combine forces with other groups. A group, here, simply refers to any number of Red teamers, i.e. ≥ 1 . After a group may or may not have joined forces with other groups, they all commence the simultaneous launch of their CyberNukes. The total damage done is $\sum_{j=1}^n a_j$.

The Red team wins if at any time $t \geq 1$, they have total CyberNuke strength greater than or equal to the CyberShield hitpoints. If it is never possible, then the Blue team wins.

If the Red teamers are unable to penetrate the network at the i^{th} second, the CyberShield hitpoints regenerate back to complete health.

Since Aryan and Dhruv have to attend another conference for 1337 h4ck3r5, they are interested in finding the minimum time it would take for one of the teams to win. If the Red team can win, your answer should be the minimum time it would take them to win. If the Red team cannot win, your answer should be -1 .

Note: Read the explanations if you have any doubts.

If you lack inspiration to read that massive blob of text above, don't worry - we did that on purpose and we too know what that feels like. But, you will often encounter problems like these in many competitive programming contests, and we would like that people new to CP get a feel for these massive statement problems too. We've tried keeping statements for other problems short.

[If you want some inspiration, click me... \(it isn't a Rickroll, pinky swear\).](#)

Input Format

The first line of the input contains three space separated integers - n , H and k .

The second line contains n space separated integers a_j - the initial CyberNuke strength of the j^{th} Red teamer.

Output Format

Output a single integer.

- If the Red Team can win, then the output must be the minimum time it takes them to win.
- If the Red Team cannot win, then the output must be -1 .

Constraints

$$1 \leq n \leq 2 \cdot 10^5$$

$$1 \leq H \leq 10^{12}$$

$$1 \leq k \leq 10^{12}$$

$$1 \leq a_j \leq 10^7$$

Sample Tests

Sample Test 1

Input

```
1 5 230 50
2 30 20 30 40 50
```

Output

```
1 2
```

Explanation

At the 1^{st} second, the following events occur:

- CyberShield hitpoints become $230 + 50 = 280$

- Let's choose Red teamers a_1 and a_5 to combine forces (the choice of a group to join forces with another group is yours to make). After this, the CyberNuke strengths become: $[60, 20, 30, 40, 100]$
- CyberNukes are launched with total damage capacity of $60 + 20 + 30 + 40 + 100 = 250$
- As the damage is lesser than the shield hitpoints, all the shields restore back to full health.

At the 2^{nd} second, the following events occur:

- CyberShield hitpoints become $280 + 50 = 330$
- Let's choose Red teamers a_4 to combine forces with the already existing group of a_1 and a_5 (the choice of a group to join forces with another group is yours to make). After this, the CyberNuke strengths become: $[90, 20, 30, 120, 150]$
- CyberNukes are launched with total damage capacity of $90 + 20 + 30 + 120 + 150 = 410$
- As the damage is greater than or equal to the shield hitpoints, the Red team wins in 2 seconds.

Sample Test 2

Input

1	3 10 20
2	3 5 7

Output

1	-1
---	----

Explanation

No matter in what order the Red teamers form groups, they can never have more CyberNuke strength than CyberShield hitpoints.

Let's take one of the cases:

At the 1^{st} second, the following events occur:

- CyberShield hitpoints become 30
- Let's say a_2 and a_3 team up. The new attack strength becomes: $[3, 10, 14]$. The total damage is 27 which isn't enough to break through the CyberShield
- The CyberShield is restored back to full health

At the 2^{nd} second, the following events occur:

- CyberShield hitpoints become 50
- Let's say a_1 joins forces with the existing group of a_2 and a_3 . The new attack strength becomes: $[9, 15, 21]$. The total damage is 45 which isn't enough to break through the CyberShield.
- The CyberShield is restored back to full health

It can easily be seen that there is no way for the Red team to increase their damage output after the first two seconds. The Blue team wins in this case.

Sample Test 3

Input

```
1 1 1295 42
2 1337
```

Output

```
1 1
```

Solution

Time Complexity: $O(n \cdot \log(n))$

Space Complexity: $O(n)$

Apologies for the long problem statement if that caused any problems for you. There are many problems with huge statements like these in competitive programming contests. Getting used to reading and navigating to the important sections of the statement is an important skill to develop.

The solution is to simply simulate what is being asked by the statement. If the Red teamers cannot defeat the Blue teamers at some time t , then it is always better for one of them to join forces with some group (as it increases the total damage output for the next second). Also, we only need to simulate for a duration of $n - 1$ seconds because at that point of time we have a single group with all n attackers. We cannot increase the attack strength any more after that.

Note that it is always better for the Red teamers to join a single group instead of having multiple groups. That way, the size of a single group is large and so their individual attack strength a_j becomes larger.

We start creating groups by merging the Red teamers with the highest attack strength together first. That way, the total increase is always the maximum possible at each second.

At the i^{th} ($1 \leq i \leq n - 1$) second, the Blue team strength is $H + i \cdot k$ and the Red team attack strength is $\sum_{j=1}^{i+1} b_j \cdot (i + 1) + \sum_{j=i+2}^n b_j$. We check if attack strength is greater than defense strength at each second. If it is greater at some second then Red Team can win, otherwise Blue Team wins.

Another approach: Note that $H + i \cdot k$ and summation of attack strength at any time i are both increasing functions. When we have functions that are monotonic, we can easily binary search for a value that satisfies a certain property.

Solution in C++:

```
1  /* Arrow */
2
3  #ifdef LOST_IN_SPACE
4  #   if   __cplusplus > 201703LL
5  #       include "lost_pch1.h" // C++20
6  #   elif __cplusplus > 201402LL
7  #       include "lost_pch2.h" // C++17
8  #   else
9  #       include "lost_pch3.h" // C++14
10 #   endif
11 #else
```

```

12 # include <bits/stdc++.h>
13 #endif
14
15 constexpr bool test_cases = false;
16
17 void solve () {
18     int64_t n, H, k;
19     std::cin >> n >> H >> k;
20
21     std::vector<int64_t> a (n);
22     int64_t strength = 0;
23     for (int i = 0; i < n; ++i) {
24         std::cin >> a[i];
25         strength += a[i];
26     }
27
28     std::sort(a.rbegin(), a.rend());
29
30     int64_t sum = a[0];
31     strength -= a[0];
32     for (int i = 1; i < n; ++i) {
33         sum += a[i];
34         strength -= a[i];
35         int64_t shield = H + k * i;
36         int64_t nuke = (i + 1) * sum + strength;
37
38         if (nuke >= shield) {
39             std::cout << i << '\n';
40             return;
41         }
42     }
43
44     // handles n = 1 case. answer is -1 if n != 1 because we already couldn't
45     // beat
46     // Blue team with all Red team members at (n - 1)'th second
47     if (n * sum >= H + k * n)
48         std::cout << n << '\n';
49     else
50         std::cout << -1 << '\n';
51 }
52
53 int main () {
54     std::ios::sync_with_stdio(false);
55     std::cin.tie(nullptr);
56     std::cout.precision(10);
57     std::cerr.precision(10);
58     std::cout << std::fixed << std::boolalpha;
59     std::cerr << std::fixed << std::boolalpha;
60
61     int cases = 1;
62     if (test_cases)
63         std::cin >> cases;
64     while (cases--)
65         solve();
66
67     return 0;
68 }

```

Solution in Python:

```
1 n, H, k = map(int, input().split())
2 a = sorted(map(int, input().split()), reverse = True)
3
4 strength = sum(a)
5 group_strength = a[0]
6 strength -= a[0]
7
8 for i in range(1, n):
9     group_strength += a[i]
10    strength -= a[i]
11
12    shield = H + i * k
13    nukes = group_strength * (i + 1) + strength
14
15    if nukes >= shield:
16        print(i)
17        exit()
18
19 if n * group_strength >= H + n * k:
20     print(n)
21 else:
22     print(-1)
```