PRACTICE   CERTIFICATION^NEW   COMPETE   JOBS   LEADERBOARD   Search   f20180084 ⌄

Distribute the Candies   🔒 locked

Problem   Submissions   Leaderboard   Discussions

Your school is holding an event today on the occasion of children's day. As a member of the school student council you have been given the responsibilty to distribute candies to the childrens who visit school for attending the event.

There are total n different types of candies available. For the ith type of candy, there are $Arr[i]$ candies of that type available in the bag. Like $Arr[2]$ candies of the type 2, $Arr[3]$ candies of type 3 ... etc. $(0 \leq i \leq n-1)$.

Now, being bored you decided to distribute the candies in the following way : Whenever a child comes to take candies, you give him x number of candies of **each** type available in the bag at that moment where $x = min(Arr[i] \mid 0 \leq i \leq n-1)$ and $x > 0$. A type of candy is available only if it's quantity is > 0.

A total of C children came to attend the festival and hence you distibuted the candies to them in the way described above. **It is gauranteed that each kid will get some candies.**

After the event, your friend came asking for the remaining candies. He says he wants *atleast* y candies from the candies left with you. Now, your task is to give him atleast y candies but should be **as low as possible** using the method described below.

Let the number of candies you'll be giving to your friend be $R = \sum max(Q_i - p, 0)$, where Q_i represents the quantity of the candy of type i remaining after you have distributed them to the children $(0 \leq i \leq n-1)$ and $(Q_i - p)$ is the quantity of candy of type i that you'll be giving to your friend.

Your task is to find a value of p so that $R = \sum max(Q_i - p, 0)$ is minimized with the constraints $y \leq R$.

Print **-1** if no value of p can be found

Refer to Sample test cases for explaination

Input Format

- **Note1:** Value can be **>=10^9**, so handle carefully (**use long long**)

- **Note2:** In C++, use fast I/O by adding `ios_base::sync_with_stdio(false); cin.tie(NULL);` as the first line inside main().

First line contains n , C and y

Second line contains n space separated integers where the ith entry represents the number of candies of type i where $0 \leq i \leq n-1$.

Constraints

- **Subtask 1 : 30 points**

1<=n<=2*10^4 ; 1<=Arr[i]<=10^7

1<=C<=10^4

1<=y<=10^9

- **Subtask 2 : 30 points**

1<=n<=10^6 ; 1<=Arr[i]<=10^4

1<=C<=10^4

1<=y<=10^6

- **Subtask 2 : 40 points**

1<=n<=10^6 ; 1<=Arr[i]<=10^10

1<=C<=5*10^5

1<=y<=10^15

Output Format

In first line, print array Arr in increasing order after candies have been distributed to k childrens.

In next line, print p and R which is to be found by you.

Sample Input 0

```
6 3 2
7 3 2 2 5 4
```

Sample Output 0

```
0 0 0 0 1 3
1 2
```

Explanation 0

For the first child, the minimum is 2 candies of type 2 and 3(Arr[2] = Arr[3] = 2), so 2 is subtracted from all. Candies available : **[5,1,0,0,3,2]**.

For the second child, the minimum is 1 candy of type 1(Arr[1] = 1), so 1 is subtracted from all available candies. Candies available : **[4,0,0,0,2,1]**

For the third child, the minimum is 1 candy of type 5(Arr[5] = 1), so 1 is subtracted from all available candies. Candies available : **[3,0,0,0,1,0]** We print the remaining array in sorted way now --> **[0,0,0,0,1,3]**

Now, the 2nd part requires us to find R and p such that $R = \sum(Q_i - p)$ is minimized.

Here array Q = [0,0,0,0,1,3] and y = 2, and R >= y

For p = 0 ;R = 0 + 0 + 0 + 0 + 1 + 3 = 4

For p = 1 ;R = 0 + 0 + 0 + 0 + 0 + 2 = 2

For p = 2; R = 0 + 0 + 0 + 0 + 0 + 1 = 1 ---> But we need R >= y

Therefore R = 2 and p = 1, which is the required answer .Here, p=0 is not an answer since we want value of R>=y along with R's value to be kept as minimum as possible .

Sample Input 1

```
4 1 5
2 5 2 3
```

Sample Output 1

```
0  0  1  3
-1
```

Explanation 1

After the first child, available candies : **[0,3,0,1]**

For y = 5, **NO** value of p satisfies the given constraints

Sample Input 2

```
5 1 5
4 3 3 5 1
```

Sample Output 2

```
0 2 2 3 4
1 7
```

Submissions: 279
Max Score: 100
Difficulty: Medium

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

C

```c
1  #include <stdio.h>
2  #include <string.h>
3  #include <math.h>
4  #include <stdlib.h>
5
6  int main() {
7
8      /* Enter your code here. Read input from STDIN. Print output to STDOUT */
9      return 0;
10 }
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code