Problem name: Balls

Language: English

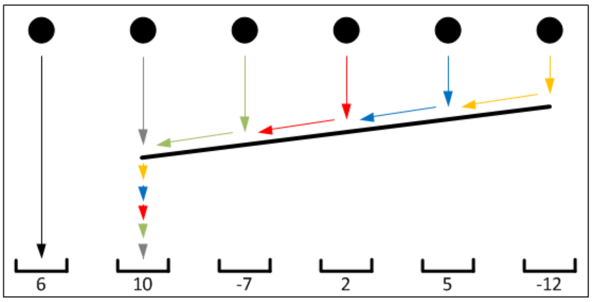
Source code:	**balls.c, balls.cpp, balls.pas**
Input / Output files:	**balls.in / balls.out**
Time limit:	**0.2 s**
Memory limit:	**256 MB**

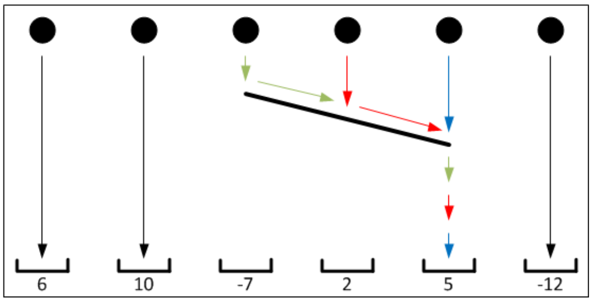
Task description

There is one standard coordinate system with *n* balls and *n* holes in it: balls are located at positions *(1, h), (2, h), …, (n, h)*, for some $h > 0$, and holes are located at *(1, 0), (2, 0), …, (n, 0)*. At a given moment, all balls start falling down (along the *y*-axis, in the negative direction). For each ball that falls into the *i*-th hole, we get c_i points. Of course, in order to avoid trivial $c_1 + c_2 + … + c_n$ score, **it is allowed to put an obstacle.**

A **left obstacle** is a segment (x_1, y_1) - (x_2, y_2) with properties: $x_1, x_2 \in N$, $1 \le x_1 < x_2 \le n$ and $0 < y_1 < y_2 < h$ (i.e. obstacle is tilted *to the left*). A **right obstacle** is a segment (x_1, y_1) - (x_2, y_2) with properties: $x_1, x_2 \in N$, $1 \le x_1 < x_2 \le n$ and $h > y_1 > y_2 > 0$ (i.e. obstacle is tilted *to the right*). If a ball, during its fall, hits an obstacle (segment), the ball temporarily stops, then slides to the obstacle's lower end and, finally, falls directly into the hole under the obstacle's lower end.

The left image shows trajectories of balls when the *left obstacle* is set, while on the right image we can see ball's trajectories when a *right obstacle* is set. Notice that when an obstacle is set, there are holes with multiple balls in them and holes with no balls in them; **however, every ball falls into one of the holes.**





The value of each hole is known. Your task is to answer the following two questions:

- What is the highest possible score you can achieve if you must set exactly one right obstacle?
- What is the highest possible score you can achieve if you must set exactly one left obstacle?

Input

The first line of input contains one integer *n* - the number of balls and holes. The second line of input contains *n* integers c_i delimited by spaces - the values of the holes, ordered from left to right.

Output

In the first line of output write a single integer – the answer to the first question. In the second line of output write a single integer – the answer to the second question. **You should use a 64-bit integer data type.**

Example

Input:	Output:

`6`
`6 10 -7 2 5 -12`
	`19`
	`56`

Example explanation

The optimal setting of right / left obstacles is shown in the images. By setting the right obstacle as shown in the right image, balls *3, 4* and *5* will fall down into hole number 5 and the score will be *6 + 10 + 5 + 5 + 5 + (-12) = 19* points. Analogously, by setting left obstacle as on the left image, we get *6 + 10 + 10 + 10 + 10 + 10 = 56* points. It is not possible to score more points with any different obstacle setting.

Constraints

- $3 \le n \le 300\,000$.
- $-10^9 \le c_i \le 10^9$.
- In 30% of test cases, it will hold $|c_i| \le 100$, for each $1 \le i \le n$.
- If you answer correct to **exactly one** question, you will get 20% of the test case value; however, you should output **two** numbers in any case.