

TASK	DVAPRAVCA	IGRACI
input	standard input	
output	standard output	
time limit (per test case)	1 second	0.4 seconds
memory limit (heap+stack)	128 MB	
points	100	100
	200	

There are points in the plane, some red some blue. The points lived in harmony for years, until the blue points went bonkers and started attacking red points. To protect themselves, the red points decided to erect two **parallel** lines such that **no blue points are contained between** the lines. All red points between the lines will be protected by them. Lines may not pass through any points, red or blue.

The red points observed that, unfortunately, not all of them can be saved in this way. Determine the **largest** number of red points that can be saved.

INPUT

The first line contains an integer N ($1 \leq N \leq 1000$), the number of points in the plane.

Each of the following N lines contains the coordinates of one point and its colour. Coordinates are pairs of integers less than 10^9 (one billion) in absolute value; the colour is 'R' or 'B'.

No three points will be collinear.

OUTPUT

Output the largest number of red points that can be protected by two parallel lines.

SCORING

In test cases worth 50% of points, N will be at most 350.

EXAMPLES

input 4 0 0 R 0 1 B 1 1 R 1 0 B output 2	input 8 2 -3 R 4 -1 R -2 0 R -3 1 B -2 3 R 1 4 R 2 1 B 0 -3 B output 3
---	---

A new computer game in the Star Expertise series is about to be released. Gamers have lined up in front of one store waiting for it to open.

While waiting, gamers exchange recent experiences from games, occasionally drifting into real-world subjects. One unavoidable topic is their computer setups. A special source of pride and arrogance is the amount of memory on the graphics card inside a computer.

Such discussions often become unpleasant so the store decided to disallow access to gamers they consider too arrogant. To ensure objectivity, they devised a simple mathematical model. When a new gamer tries to enter the line, his graphics card is compared with the cards of players already in line, using a simple criterion: the more memory his graphics card has, the better it is. Each of the quotients (the memory of the new gamer divided by the memory of one already in line) is rounded **down**, and then the total **arrogance** of the new gamer is estimated to be the **sum of all these quotients**.

For example, suppose there already three players in line, with 3, 1 and 2 megabytes on their graphics cards. A new gamer with 3 megabytes will have arrogance $1+3+1=5$.

The store management will not allow a gamer to enter the line if his arrogance is greater than the number of people already in line. In the above example, the new gamer with 3 megabytes of memory and arrogance 5 would be denied access, but a gamer with 2 megabytes would be allowed, since his arrogance would be $0+2+1=3$ (which is less than or equal to 3, the number of people already in line).

Write a program that, **given the calculated arrogances** of gamers in line, **finds one possible sequence of memory amounts** on their graphics cards.

INPUT

The first line contains an integer N ($1 \leq N \leq 100\,000$), the number of players in line.

The second line contains N non-negative integers, the arrogances of players in line. The arrogances will be given in order in which the gamers arrived. Additionally, the arrogance of gamer k (starting from 1) will be less than k .

OUTPUT

Output N integers on a single line, for each gamer the amount of memory on his graphics card, in the same order in which the gamers' arrogances were given. The amount of memory must be an **integer less than 10^9** . The solution may not be unique, but will always exist.

EXAMPLES

input 4 0 0 1 3 output 6 2 3 4	input 6 0 1 0 2 0 3 output 7 7 3 6 2 5
--	--