Robot

HackerRank

You have two arrays of integers, $V = \{V_1, V_2, \ldots, V_N\}$ and $P = \{P_1, P_2, \ldots, P_N\}$, where both have N number of elements. Consider the following function:

```
score = 0

int Go(step, energy) {
    if (step == N) {
        score += V[step];
        return (score);
    }
    else {
        int way = random(1, 2);
        if (way == 1) {
            score += V[step];
        }
        else {
            energy = P[step];
        }
        if (energy > 0) {
            Go(step + 1, energy - 1);
        }
        else {
            KillTheWorld();
        }
    }
}
```

What is the maximum possible value of score that we can get in the end, if we call $Go(1, 0)$?.
Note that the function should never invoke *KillTheWorld* function. And $random(1, 2)$ generates a random integer from set *[1, 2]*.
It is guaranteed there will be a solution that *wont* kill the world.

Input Format

The first line contains an integer N. Each of the following N lines contains a pair of integers. The i-th line contains a pair of numbers, V_i, P_i, separated by space.

Constraints

$1 \leq N \leq 5 \times 10^5$
$0 \leq V_i \leq 10^9$
$0 \leq P_i \leq 10^5$

Output Format

Derive the maximum score given by `return (score);`.

Sample Input

```
4
4 2
0 2
```

```
4 0
3 4
```

Sample Output

```
7
```

Explanation

In the best case, the first and second function call in Go variable way will take value 2, while in the other calls it will be equal to 1 then the final score will be equal to the value of 7.