

След поредното онлайн контролно при  $C^2$ ,  $C^2$  е обещал да провери контролните на всички веднага. Дори е решил, че ако някой предаде по-рано, то  $C^2$  веднага ще започне да проверява, за да не губи време.  $C^2$  обаче иска да проверява така че средното време за чакане за проверка на всички да е минимално, за да не се сърди никой, че е останал последен.  $C^2$  проверява 1 ред код за 1 секунда.

Намерете минималното средно време, което трябва да чака един студент, за да му бъде проверено контролното.

Забележка:  $C^2$  не може да вижда бъдещето, така че ако не проверява на никого и някой студент предаде контролно  $C^2$  започва веднага да проверява. Това предполага че може в следващият момент да бъде предадено по-удачно за проверяване контролно, но  $C^2$  няма как да знае за него.

## Input Format

На първият ред ще получите едно число  $N$  - броя на студентите.

На следващите  $N$  реда ще получите по 2 числа:

$submitTime_i$   $linesCode_i$  за всеки студент, където

- $submitTime_i$  е времето когато  $i$ -тия студент предава контролното, а
- $linesCode_i$  е колко реда код е предал

Забележка: Студентите на входа не са подредени.

## Constraints

$$1 \leq N \leq 250,000$$

$$0 \leq submitTime_i \leq 10^9$$

$$0 \leq linesCode_i \leq 10^9$$

## Output Format

На 1 ред изведете единствено число: Минималното средно време, което всеки студент трябва да чака, за да му бъде проверено контролното. Отговорът закръглете на долу до цяло число.

## Sample Input 0

```
3
10 5
14 10
15 5
```

## Sample Output 0

```
8
```

## Explanation 0

Имаме трима студенти които предават следните контролни:

- Студент 1: 5 реда код на 10 секунда
- Студент 2: 10 реда код на 14 секунда
- Студент 3: 5 реда код на 15 секунда

Ако Ч<sup>2</sup> проверява под ред то първият студент ще чака 5 секунди, вторият ще чака 1+10=11 секунди и третият ще чака 10+5=15 секунди. Средното време за чакане е  $(5+11+15)/3 = 10$  (закръглено на долу).

Но има по-добро средно време: Ако след първият студент Ч<sup>2</sup> провери третия, то времената за чакане ще са: 5 за първия, 5 за третия, 6+10 за втория. Тогава средното време за чакане ще е  $(5+5+16)/3 = 8$  (закръглено на долу).

След като Ч<sup>2</sup> е проверил на първия предал, има двама чакащи - втория и третия. При наивното изчисление Ч<sup>2</sup> проверява първо на втория, после на третия. При второто изчисление Ч<sup>2</sup> проверява първо на третия, защото така ще минимизира средното време за чакане.

## Sample Input 1

```
3
0 5
10 20
11 4
```

## Sample Output 1

```
16
```

## Explanation 1

Първият студент предава веднага 5 реда код. Тъй като Ч<sup>2</sup> няма работа в момента - почва да проверява и след 5 секунди е приключил (Студент 1 чака 5 секунди). След това до 10та секунда Ч<sup>2</sup> отново няма какво да проверява и чака (това чакане не ни засяга).

На 10та секунда предава студент с 20 реда код и тъй като Ч<sup>2</sup> няма други контролни веднага почва да го проверява.

На 11та секунда предава студент с 4 реда код, но Ч<sup>2</sup> вече проверява друго контролно и не може да прекъсне проверката. Това води до неоптимален отговор, но това е неизбежно защото Ч<sup>2</sup> не вижда бъдещето.

На края: студент 1 чака 5 секунди, студент 2 чака 20 секунди, студент 3 чака 19+4 секунди. Следователно средното време за чакане е  $(5+20+23)/3 = 16$  секунди

## Sample Input 2

```
3
0 10
0 15
11 1
```

## Sample Output 2

16

## Explanation 2

UPDATE! Условието не е напълно коректно! При няколко налични контролни за проверка трябва да изберем "най-бързата" за проверяване.