

Прави

След като прочел във Facebook, че Клод е приключил с програмирането, Фики е решил да започне да се занимава с физика.

Той мисли, че е открил нова физична формула за определяне на това колко килограма агнешко може да изяде спрямо времето на последното му хранене. Формулата е от следния вид: $y = m \cdot x + b$, където y е количеството агнешко, което Фики може да изяде, а x е времето от последното му хранене. m и b са константи, които задават наклона и отстоянието на правата по y , съответно.

Фики е събрал n данни, за да потвърди своя пробив. Данните са зададени като (x_i, y_i, d_i) , където x_i е времето спрямо последното му хранене, y_i е измерването на везната за количеството изядено агнешко и d_i е неточността на везната, с която е направено измерването. Това означава, че истинското количество агнешко, което е изял, е в интервала $[y_i - d_i, y_i + d_i]$, но той не знае точното количество. За целите на задачата m може да е отрицателно число и y_i може да е по-малко от d_i .

Може да представим i -тата данна като вертикална отсечка, чиито краища са точките $(x_i, y_i - d_i)$ и $(x_i, y_i + d_i)$.

Фики иска да знае дали съществува **не**вертикална права, която минава през всички тези отсечки. Казано по друг начин, правата трябва да пресича всяка отсечка.

Ако такава права не съществува, трябва да съобщите това. В противен случай трябва да намерите минималния и максималния възможен наклон на такава права.

Обърнете внимание, че правите с минималния и максималния наклон може да нямат едно и също отстояние по y .

Вход

На първия ред има положително цяло число n .

След това има n реда с едно цяло число x_i и две дробни числа y_i и d_i . Дробните числа имат най-много 9 цифри след десетичната точка.

Гарантирано е, че всички x_i са уникални.

Исход

Ако не съществува такъв наклон, отпечатайте -1 . В противен случай отпечатайте 2 дробни числа: минималния и максималния възможен наклон.

За двете числа абсолютна или релативна грешка до 10^{-6} е позволена. Формално, ако сте отпечатали x' и очакваният отговор е x , то вашият отговор се счита за верен при условие че:

$$\frac{|x-x'|}{\max(1,|x|)} \leq 10^{-6}.$$

Съвети за имплементацията

За по-точни дробни сметки използват типа `long double` в C++. Освен това, когато отпечатвате дробни числа е добре да зададете точност. По-долу има извадка от код, за начин по който това може да се направи.

```
long double val = 10;
std::cout << std::fixed << std::setprecision(9) << val << "\n";
```

Уверете се, че имате `include` на `<iomanip>` хедъра, когато използвате `setprecision`.

Ограничения

Гарантирано е, че ако съществува валидна права, то разликата на максималния и минималния наклон е поне 0.1. Ако не съществува валидна права, то няма валидна права дори всяко d_i да се увеличи с 10^{-3} . Също така е гарантирано, че за всяко валидно m , $|m| \leq 10^3$.

- $2 \leq n \leq 2 \cdot 10^5$
- $0 \leq x_i \leq 10^6$
- $0 \leq y_i \leq 10^9$
- $0 < d_i \leq 10^9$

Подзадачи

Подзадача 1 (5 точки)

$$n = 2$$

Подзадача 2 (7 точки)

Гарантирано е, че има права с наклон 1, която минава през всички отсечки.

Подзадача 3 (19 точки)

$$n \leq 100$$

Подзадача 4 (29 точки)

$$n \leq 2000$$

Подзадача 5 (40 точки)

Няма допълнителни ограничения.

Пример

Вход 1

```
2
0 0 1
10 10 1
```

Изход 1

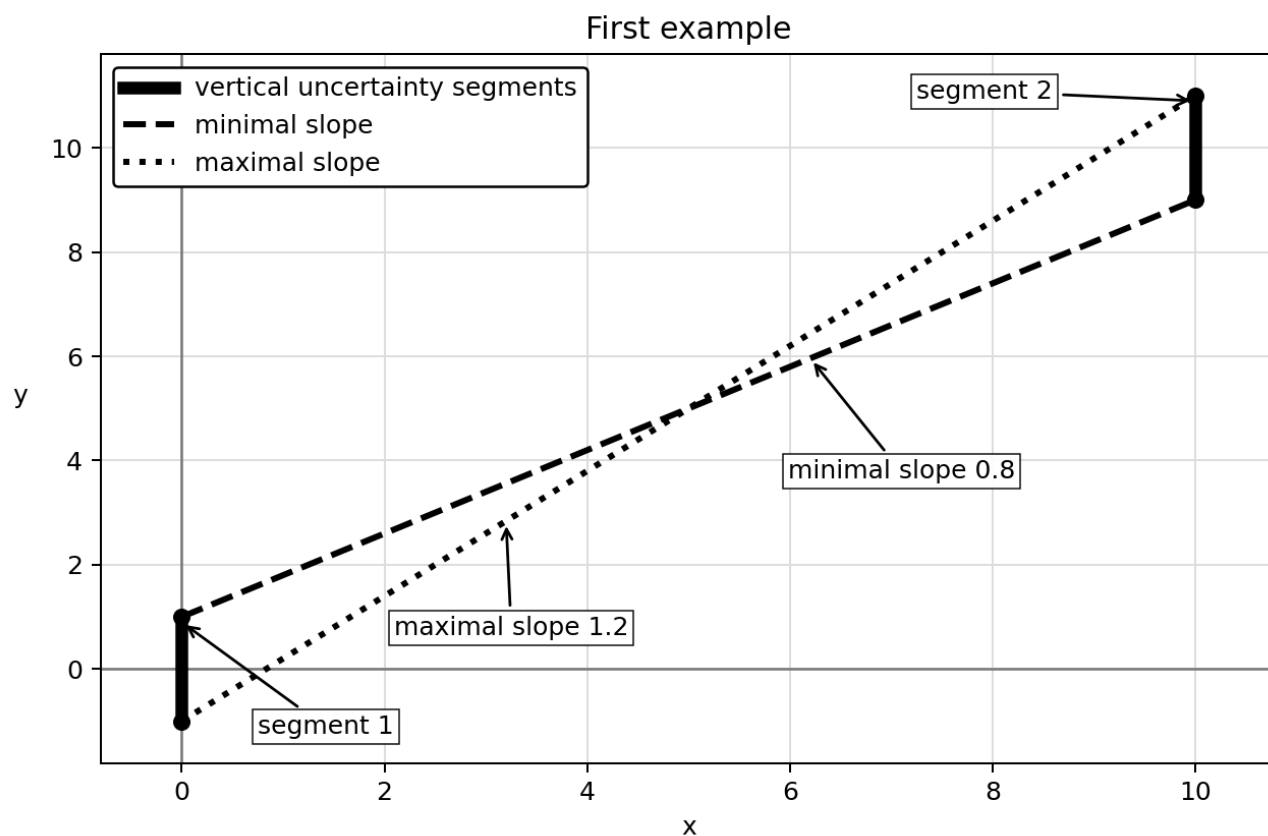
```
0.800000000000 1.200000000000
```

Обяснение 1

Първата отсечка е от точка $(0, -1)$ до $(0, 1)$, а втората отсечка е от точка $(10, 9)$ до $(10, 11)$.

Най-малкият възможен наклон се постига като се свържат точките $(0, 1)$ с $(10, 9)$, постигайки наклон 0.8. Най-големият възможен наклон се постига като се свържат точките $(0, -1)$ с $(10, 11)$, постигайки наклон 1.2.

Визуализация на примера:



Вход 2

```
3
0 0 0.7
2 0 1.5
1 99.3 1
```

Изход 2

```
-1
```

Обяснение 2

Няма **не**вертикални прави, които пресичат всички три отсечки, защото средната отсечка е твърде нагоре спрямо другите две външни отсечки.