Quả bài này khó quá, câu A thốn vcl. Éo ăn nổi bài nào

1. **Integer Moves**

Chạy từ (0,0) -> (x,y) nhưng mà chỉ được chạy qua (x1,y1) -> (x2, y2) là số nguyên.

Tìm **min** số bước đi.

(0,0) -> (a,b) -> (c,d) -> … -> (x,y)

**Hint:** x,y <= 50, như Brute Force 2^50 maybe 😊

**Khó:** Éo biết có bao nhiêu bộ 3 Pitago éo có giới hạn tới 50, có thể cộng trừ, nên ko chỉ từ 0 -> 50, nó chạy xuống 0 -> -40 -> 50. Éo xài dynamic programming đc, cay 😊

Nghiệm của bộ 3 Pitago: a^2 + b^2 = c^2 => a = x^2 – y^2, b = 2xy

Xong 0 +- a1 +- a2 +- … = x, và 0 +- b1 +- b2 … = y. Kiểm soát 2 số, kiểm soát nghiệm Pitago, kiểm soát dấu cộng trừ. Thua !!!! .\_.

**Answer:** Không ngoài dự đoán thì bài này dễ vcl, chỉ có 3 giá trị 0,1,2. Mình đoán đúng nhưng cái sqrt() -> Double, ko thể lường trước được. Hảo vcl 😊

(x,y) = (0,0) -> 0

(x,y): x^2 + y ^2 = z^2 -> 1

Chứng minh: (x,y): x^2 + y^2 ko là SCP thì tồn tại (a,b):

(0,0) -> (a,b) ->(x,y) để 2 cái này là số nguyên. (Khó vãi loz nhưng chắc vị thần nào đó sẽ làm ra)

1. **XY Sequence**

Cho n,B,x,y. Tìm chuỗi a0, a1, .. an: a0 = 0, ai = a(i-1) + x or a(i-1) -y + a(i) <= B

Tìm **max** của sigma chuỗi

**Hint:**  n < 2\*10^5, B,x,y < 10^9. Lại là cộng trừ vô tội vạ

**Idea:** Thường thì cứ cộng x cho đến khi đạt giới hạn B rồi trừ 1 phát y rồi cộng tiếp x => Naïve nghe có vẻ max+ Dễ thế thì Failed

Còn vài trường hợp sâu hơn, y >> x hay x >> y. Lúc đó, ta có thể **hy sinh** 1 lần cộng để được nhiều lần cộng hơn ở phía sau.

1 idea khác. Khi sigma lại, thì dấu + ở lần chọn đầu tiên sẽ được lặp lại n lần, quyết định lần thứ 2 sẽ lặp lại n-1, lần thứ n lặp lại 1 lần. Vậy nên tùy vào x,y sẽ so sánh thử độ quan trọng sẽ chọn +, - 1 cách hợp lý.

**Answer:** Do chưa dùng long long thôi

1. **Bracket Sequence Deletion**

Cho chuỗi 2 dấu ( ), 1 lần xóa thì **shortest prefix** mà **good** của string.

Good = Regular (Dùng stack như thường) + Palindrome len >= 2 (đối xứng)

Làm cho đến khi ko tìm ra Good. Tìm số bước.

**Hint:** Stack thì ez, quét đến lúc đúng là shorest của 1 cái.

Còn lại thì thấm: Tìm **longest prefix palindrome** của string. **Hard** các kiểu

<https://www.geeksforgeeks.org/print-the-longest-palindromic-prefix-of-a-given-string/>

<https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/> (Bonus)

Có min 2 cái, xóa đi và làm đến khi not good => **Done**

**Answer:** Vẫn hơi lú

1. **For Gamers. By Gamers**

Có c tiền mua 1 loại tướng, n con tướng: Tiền 1 con, damage, máu. Có m con quái: Dame, máu. Hỏi đánh thắng con quái thứ m thì tốn ít nhất nhiêu tiền? Đánh thắng: Mình giết trước nó trước khi nó giết 1 con bên mình: **Dame(Nhiều con cộng dồn vào) \* Heal > Dame quái \* Heal quái**

1. **Star MST**
2. **Words on tree**