# 19. Mở rộng thành phố (P19.\*) - [424B](http://www.codeforces.com/contest/424/problem/B)

Các nhà lãnh đạo của thành phố Tomsk tin tưởng rằng trong thời gian tới thành phố này sẽ trở thành siêu đô thị (có dân số từ 1 triệu người trở lên). Tuy nhiên thay vì đợi dân cư phát triển một cách tự nhiên, ban lãnh đạo lại quyết định gom các điểm dân cư nằm rải rác trong vùng vào thành phố để tạo ra một siêu đô thị ngay lập tức.

Vị trí thành phố Tomsk đặt tại (0,0). Có điểm dân cư xung quanh, điểm dân cư thứ có thể được mô tả như là một điểm trên mặt phẳng tọa độ với dân số . Theo quyết định, tất cả các điểm dân cư nằm cách (0,0) không quá sẽ được nhập vào Tomsk.

Hãy xác định bán kính nhỏ nhất để nếu gom các điểm dân cư theo qui tắc nói trên thu được một siêu đô thị.

*Input:*

* Dòng thứ nhất ghi hai số nguyên dương ( là số điểm dân cư xung quanh và dân só hiện thời của Tomsk
* dòng tiếp theo, dòng thứ ghi ba số nguyên (, các tọa độ có giá trị tuyệt đối không vượt quá 104. Giả thiết rằng không có điểm dân cư nào đặt tại (0,0).

*Output:* Ghi -1 nếu như không lập đươc siêu đô thị. Trường hợp ngược lại, ghi bán kính tối thiểu để mở rộng thành phố với độ chính xác đến .

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **4 999998**  **1 1 1**  **2 2 1**  **3 3 1**  **2 -2 1** | **2.8284271** |

# 20. Rót nước (P20.\*) - [426A](http://www.codeforces.com/contest/426/problem/A)

Sereja có một cái bình rỗng và cốc nước đầy. Anh ta muốn có cốc nước rỗng để dùng vào việc khác. Để làm điều này Sereja sẽ thực hiện lần thao tác sau: lấy một cốc nước đầy và rót toàn bộ số nước trong đó vào bình. Sereja muốn rằng bình không bị tràn nước ra ngoài. Hỏi rằng có thể làm được điều đó hay không?

*Input:* Gồm nhiều bộ dữ liệu. Mỗi bộ dữ liệu gồm hai dòng theo qui cách sau

* Dòng đầu tiên ghi hai số nguyên dương - số cốc nước đầy và dung tích của bình nước (
* Dòng thứ hai chứa số nguyên dung tích của các cốc nước

*Output:* Với mỗi bộ dữ liệu, in ra 'YES' hoặc 'NO' nếu có thể/không thể có được cốc rỗng

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **3 4**  **1 1 1**  **3 4**  **3 1 3**  **3 4**  **4 4 4** | **YES**  **YES**  **NO** |

# 21. Phép đối xứng ma trận (P21.\*) - [426B](http://www.codeforces.com/contest/426/problem/B)

Giả sử có ma trận kích thước . Khi đó ma trận đối xứng gương của là ma trận kích thước được xây dựng theo qui tắc:

* Nửa trên của ma trận chính là ma trận
* Nửa dưới của ma trận (từ hàng đến hàng ) là các hàng đối xứng vởi nửa bên trên (đường đối xứng nằm giữa hàng và hàng )

Sereja có ma trận kích thước . Anh ta muốn tìm ma trận sao cho có thể thực hiện một số lần đối xứng (có thể là 0) để nhận được ma trận . Nếu có nhiều ma trận như vậy thì số hàng nhỏ nhất của ma trận bằng bao nhiêu?

*Input:*

* Dòng đầu tiên ghi hai số nguyên dương
* dòng tiếp theo, dòng thứ ghi số nguyên mô tả hàng thứ của ma trận (

*Output:* Một số nguyên là số hàng nhỏ nhất của ma trận

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **4 3**  **0 0 1**  **1 1 0**  **1 1 0**  **0 0 1** | **2** |

# 22. Cảnh sát tân binh (P22.\*) - [427A](http://www.codeforces.com/contest/427/problem/A)

Sở cảnh sát trong thành phố của bạn bắt đầu công việc điều tra tội phạm. Do đó họ tiến hành tuyển dụng các nhân viên điều tra mới. Trong khi đó, tội phạm vẫn liên tiếp xuất hiện trong thành phố và vì là các nhân viên mới nên mỗi người chỉ tiến hành điều tra một tội phạm duy nhất trong thời gian làm việc của mình. Nếu một tội phạm xuất hiện mà không có nhân viên mới nào rỗi việc thì tội phạm này không bị điều tra.

Cho danh sách các sự kiện xuất hiện tội phạm và tuyển nhân viên mới theo thứ tự thời gian. Hãy xác định xem có bao nhiêu tội phạm không bị điều tra?.

*Input:*

* Dòng đầu tiên ghi số nguyên dương () là số lượng các sự kiện.
* Dòng tiếp theo ghi số nguyên là các sự kiện xảy ra theo trình tự thời gian. Nếu một tội phạm mới xuất hiện, nó là số -1. Trường hợp ngược lại, đây là sự kiện tuyển nhân viên điều tra mới được thể hiện bằng một số nguyên dương - số nhân viên mới được tuyển dụng cùng thời điểm (không quá 10 nhân viên một lần)

*Output:* In ra một số nguyên là số lượng tội phạm không được điều tra.

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **3**  **-1 -1 1** | **2** |
| **11**  **-1 -1 2 -1 -1 -1 -1 -1 -1 -1 -1** | **8** |

# 23. Chuyển tù nhân (P23.\*) - [427B](http://www.codeforces.com/contest/427/problem/B)

Nhà tù của thành phố có tù nhân, con số này vượt quá sức chứa của nhà tù nên cần phải chuyển một số tù nhân sang các trại giam khác.

Để làm điều này, người ta xếp tù nhân thành một hàng. Trên ngực mỗi người đeo một tấm biển ghi một con số - mức độ nguy hiểm của tội danh mà anh (cô) ta phạm phải. Con số càng lớn thì mức độ nguy hiểm càng cao.

Ông thị trưởng yêu cầu bạn phải chọn ra tù nhân để chuyển trại giam. Có hai qui tắc ông ta yêu cầu đối với bạn là:

* Bạn phải chọn ra dãy có tù nhân liên tiếp
* Bất kỳ tù nhân nào được chọn để chuyển trại phải có mức độ nguy hiểm không vượt quá . Điều này đảm bảo cho tính an toàn của việc chuyển tù nhân

Hỏi rằng bạn có bao nhiêu cách để thực hiện nhiệm vụ trên?

*Input:*

* Dòng đầu tiên ghi 3 số nguyên ()
* Dòng tiếp theo ghi số nguyên, số thứ là mức độ nguy hiểm của tù nhân đứng ở vị trí thứ . Mức độ nguy hiểm là một số nguyên không âm không vượt quá 109.

*Output:* Một số nguyên duy nhất là số cách chuyển tù nhân

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **4 3 3**  **2 3 1 1** | **2** |
| **11 4 2**  **2 2 0 7 3 2 2 4 9 1 4** | **6** |

# 24. Điểm và đoạn thẳng (P24.\*) - [430A](http://www.codeforces.com/contest/430/problem/A)

Iahub không được chuẩn bị tốt để làm các bài lập trình hình học. Tuy nhiên nghe đồn rằng trong kỳ thi IOI năm nay sẽ có bài thi về hình học nên anh ta quyết định đầu tư thời gian nghiên cứu phần này.

Iahub vẽ điểm phân biệt và đoạn thẳng trên trục tọa độ OX. Mỗi điểm được vẽ bằng một trong hai màu đỏ hoặc xanh. Một cách tô màu được gọi là tốt nếu như với bất kỳ đoạn thẳng nào đã vẽ chênh lệch về số lượng điểm đỏ và số lượng điểm xanh nằm trong nó không quá 1 (ta nói điểm nằm trong đoạn nếu như )

*Input:*

* Dòng đầu tiên ghi hai số nguyên và
* Dòng thứ hai ghi số nguyên - tọa độ các điểm tô màu.
* dòng tiếp theo, dòng thứ ghi hai số nguyên thể hiện hai đầu mút của đoạn thẳng thứ .

*Output:* Nếu không có cách tô màu hợp lý thì ghi số -1. Trường hợp ngược lại ghi số nguyên 0 hoặc 1. Số thứ thể hiện màu của điểm thứ ( - màu đỏ, 1- màu xanh). Nếu như có nhiều phương án tô màu thì bạn chỉ cần in một phương án tùy ý

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **3 4**  **1 2 3**  **1 2**  **2 3**  **5 6**  **2 2** | **1 0 1** |

# 25. Trò chơi bóng (P25.\*) - [430B](http://www.codeforces.com/contest/430/problem/B)

Có quả bóng xếp thành một hàng. Mỗi quả bóng có một màu riêng biệt trong số màu và không có ba quả bóng liên tiếp nào cùng màu. Iahub có một quả bóng với màu . Anh ta có thể chèn quả bóng này vào một vị trí bất kỳ nào đó trong hàng (chèn giữa hai quả bóng khác). Tại mọi thời điểm nếu như có ít nhất ba quả bóng liên tiếp cùng màu thì các quả bóng này sẽ tự hủy và các quả bóng còn lại sẽ dồn lại tạo thành hàng liên tiếp, quá trình tự hủy sẽ tiếp tục xảy ra cho đến khi không còn quả bóng nào hoặc không có ba quả bóng nào cùng màu liên tiếp.

Ví dụ có 6 quả bóng với màu [đen, đen, trắng, trắng, đen, đen] và trên tay của Iahub có quả bóng màu trắng. Nếu anh ta chèn quả bóng màu trắng này vào giữa hai quả bóng màu trắng thì sẽ có 3 quả bóng màu trắng liên tiếp nên chúng sẽ bị hủy. Bốn quả bóng màu đen bị dồn lại thành một hàng nên cả bón quả này bị bủy. Tổng cộng có 6 quả bóng bị hủy (không tính quả bóng trên tay Iahub.

Iahub muốn hủy nhiều quả bóng nhất có thể. Bạn được cho màu các quả bóng trong hàng và màu của quả bóng trên tay Iahub. Viết chương trình tính xem Iahub có thể hủy tối đa bao nhiêu quả bóng?

*Input:*

* Dòng đầu tiên ghi ba số nguyên (, , )
* Dòng thứ hai ghi số nguyên ) là màu của các quả bóng trong hàng

Dữ liệu đảm bảo rằng không có ba quả bóng cùng màu liên tiếp trong hàng.

*Output:* In ra một số nguyên - số quả bóng nhiều nhất bị hủy

*Example:*

|  |  |
| --- | --- |
| **Input** | **Output** |
| **6 2 2**  **1 1 2 2 1 1** | **6** |