# Bài 1. TÌM NGƯỜI QUEN Tên file: PAIRS.CPP

N người đang chờ đợi xếp hàng để vào rạp chiếu phim. Mọi người chán nản vì phải chờ đợi, họ quay lại và tìm kiếm một ai đó quen thuộc trong hàng người. Hai người A và B đứng trong hàng có thể nhìn thấy nhau nếu họ đang sát nhau hoặc tất cả những người giữa họ có chiều cao thấp hơn 2 người đó.

**YÊU CẦU:** Viết một chương trình để xác định số lượng các cặp người mà có thể nhìn thấy nhau.

## INPUT: PAIRS.INP

* Dòng đầu tiên chứa một số nguyên N (1 ≤ N ≤ 500.000), số người đứng xếp hàng.
* Dòng 2 chứa N số nguyên là chiều cao của một người tính theo nanomet. Tất cả mọi người sẽ thấp hơn 231 nanomet.
* Chiều cao được đưa ra theo thứ tự mà mọi người đang đứng xếp hàng.

## OUTPUT: PAIRS.OUT:

* Số S là số lượng cặp người nhìn thấy nhau.

*Ví dụ:*

|  |  |
| --- | --- |
| **PAIRS.INP** | **PAIRS.OUT** |
| 6  2 1 4 3 6 5 | 7 |
| 5  2 2 2 2 2 | 4 |

# Bài 2. HÌNH CHỮ NHẬT LỚN NHẤT Tên file: RECT.CPP

Cho một bảng hình chữ nhật kích thước 𝑚 × 𝑛 được chia thành lưới ô vuông đơn vị 𝑚 hàng, 𝑛 cột. Các hàng được đánh số từ 1 tới 𝑚 theo thứ tự từ trên xuống dưới và các cột được đánh số từ 1 tới 𝑛 theo thứ tự từ trái qua phải.

Người ta tiến hành tô màu các ô của bảng theo từng cột: Các ô trên mỗi cột 𝑗 sẽ được tô từ trên xuống dưới: ℎ𝑗 ô màu vàng tiếp đến là 𝑚 − ℎ𝑗 ô màu xanh. Như vậy tình trạng màu trên bảng hoàn toàn xác định nếu ta biết được số hàng 𝑚, số cột 𝑛 và các số nguyên ℎ*1*, ℎ2, … , ℎ𝑛.

Hãy xác định một hình chữ nhật gồm các ô trong bảng đã cho thỏa mãn các yêu cầu sau:

* Có cạnh song song với cạnh bảng
* Đơn sắc (chỉ gồm các ô vàng hoặc chỉ gồm các ô xanh)
* Diện tích lớn nhất có thể

Chart, histogram

Description automatically generated

## INPUT: RECT.INP

* Dòng 1: Chứa hai số nguyên dương 𝑚, 𝑛 (𝑚, 𝑛 ≤ 106)
* Dòng 2: Chứa 𝑛 số nguyên ℎ*1*, ℎ2, … , ℎ𝑛 (∀𝑗: 0 ≤ ℎ𝑗 ≤ 𝑚)

## OUTPUT: RECT.OUT

* Một số nguyên duy nhất là diện tích hình chữ nhật tìm được

Ví dụ:

|  |  |
| --- | --- |
| **RECT.INP** | **RECT.OUT** |
| 5 9  1 3 4 4 5 4 4 3 1 | 21 |

# Bài 3: Harry Potter và khu vườn diệu kì Tên file: fgarden.cpp

Bộ ba Harry, Ron và Hermione đến thăm vườn nhà bác Hagrid. Trong khu vườn có n cây mận gai nối tiếp nhau, mỗi cây có một độ cao là . Một đoạn các cây mận gai liên tiếp được gọi là diệu kì nếu chúng thỏa mãn điều kiện sau: với mọi cặp thỏa mãn thì . Để chống lại Voldemort, bác Hagrid phải chọn ra một đoạn các cây mận gai dài nhất mà có tính chất diệu kì.

**Yêu cầu**: Cho n cây mận gai cùng độ cao của chúng và số nguyên không âm t, hãy tìm đoạn dài nhất mà có tính chất diệu kì.

## Input: fgarden.cpp

* Dòng đầu chứa số nguyên dương và số nguyên không âm
* Dòng thứ hai chứa số nguyên tương ứng với độ cao của từng cây.

## Output: fgarden.cpp

* Một dòng duy nhất là kết quả của bài toán.

|  |  |
| --- | --- |
| **Fgarden.inp** | **Fgarden.out** |
| 9 3  5 1 3 5 8 6 6 9 10 | 4 |

Với mọi test

* **Subtask 1:**
* **Subtask 2:**
* **Subtask 3:**
* **Subtask 4:**

# Bài 4. Nước đọng Tên file: WATER.CPP

Năm 2099, tình trạng ngập lụt trong thành phố trở nên nghiêm trọng hơn. Vì vậy, mọi người quyết định xây dựng hệ thống mái che cho toàn thành phố.

Mái che có bề rộng là ***N***, được chia làm ***N*** phần có độ dài như nhau. Độ cao của mỗi phần là ***h1, h2, ..., hN***. Khi trời mưa, một phần nước sẽ đọng lại trên mái và một phần sẽ thoát ra ngoài theo hai bên trái và phải của mái che. Do đó, thành phố sẽ không phải chịu cảnh mưa lụt như trước.

Nhằm mục đích bảo trì mái che, bạn cần viết chương trình tính lượng nước lớn nhất có thể đọng lại trên mái che.

## Input: WATER.INP

* Dòng đầu ghi số N (1 ≤ N ≤ 105)
* Dòng sau ghi N số tự nhiên h1, h2, ..., hn. (1 ≤ hi ≤ 105)

## Output: WATER.OUT

* Gồm một số duy nhất thể hiện lượng nước tìm được.

Ví dụ:

|  |  |
| --- | --- |
| **WATER.INP** | **WATER.OUT** |
| 5  1 3 1 2 3 | 3 |

# Bài 5. Hình chữ nhật con tên file: MAXREC.CPP

Cho một lươi hình chữ nhật kích thước M x N được chia thành M hàng và N cột. Mỗi một ô vuông của lưới ghi hoặc số 1 hoặc số 0.

Hãy tìm hình chữ nhật con trên lưới trên gồm một số nguyên lần các ô vuông, có các cạnh song song với các cạnh của lưới hình chữ nhật, chứa toàn số 1 và có diện tích lớn nhất.

## Input: MAXREC.INP

* Dòng đầu tiên ghi hai số nguyên M, N ()
* M dòng tiếp theo, mỗi dòng ghi N số 0/1 mô ta lưới chữ nhật

## Output: MAXREC.OUT

* Một số nguyên duy nhất là diện tích (số ô vuông) lớn nhất của hình chữ nhật chứa toàn số 1

Ví dụ:

|  |  |  |
| --- | --- | --- |
| **MAXREC.INP** | **MAXREC.OUT** |  |
| 4 5  0 0 1 1 1  0 1 1 1 1  1 1 1 1 1  0 0 0 0 0 | 9 |  |

# Bài 6. Tổng đoạn Tên file: SUMLR.CPP

Cho dãy số và 2 số nguyên dương . Tìm dãy con liên tiếp thỏa mãn:

* Số lượng phần tử nằm trong đoạn
* Tổng các phần tử đạt giá trị lớn nhất.

## Dữ liệu vào: SUMLR.INP

* Dòng đầu tiên ghi 3 số nguyên dương
* Dòng thứ 2 chứa số nguyên xác định dãy số

## Kết quả: SUMLR.OUT

* Một số nguyên duy nhất là tổng lớn nhất tìm được.

***Ví dụ:***

|  |  |
| --- | --- |
| **SUMLR.INP** | **SUMLR.OUT** |
| 9 2 3  40 -40 0 3 -5 0 3 0 1 | 4 |

**Ràng buộc:**

* số test có
* số test khác có
* số test còn lại có

# Bài 7. THẰNG BỜM VÀ PHÚ ÔNG Tên file: BOTTLES.CPP

Bờm thắng phú ông trong một cuộc đánh cược và buộc phú ông phải đãi rượu. Phú ông bèn bày ra một dãy *n* chai chứa đầy rượu, và nói với Bờm rằng có thể uống bao nhiêu tuỳ ý, nhưng đã chọn chai nào thì phải uống hết và không được uống ở *k* chai liền nhau bởi đó là điều xui xẻo.

Bạn hãy chỉ cho Bờm cách uống được nhiều rượu nhất.

## INPUT: BOTTLES.INP

* Dòng 1 chứa hai số nguyên *n* và *k* (*1 ≤ n ≤ 4\*105, 2 ≤ k ≤ n*)
* Dòng 2 chứa các số nguyên dương (*≤ 109*) là dung tích của các chai rượu phú ông bày ra, theo thứ tự liệt kê từ chai thứ nhất tới chai thứ *n*.

## OUTPUT: BOTTLES.OUT

* Dòng 1 ghi số chai được chọn và lượng rượu tối đa có thể uống.
* Dòng 2 ghi chỉ số của các chai được chọn theo thứ tự tăng dần.

*Ví dụ:*

|  |  |
| --- | --- |
| **BOTTLES.INP** | **BOTTLES.OUT** |
| 6 3  6 10 10 13 10 10 | 4 40  2 3 5 6 |

# Bài 8. Giai đoạn vui vẻ Tên file: feelgood.cpp

Alex là một bác sĩ tâm lý học rất giỏi tại Trâu Quỳ do Nga cử sang. Anh đang nghiên cứu về các phương pháp làm cho bệnh nhân cảm thấy vui vẻ hơn. Trong phương pháp này, anh ấy cần đánh giá ảnh hưởng của các hoạt động đến trạng thái vui vẻ của bệnh nhân.

Bệnh nhân sẽ được tham gia một dãy hoạt động nào đó. Các y sĩ sẽ ghi lại giá trị cảm xúc của bệnh nhân sau khi tham gia xong. Nhìn vào bảng thống kê này, bác sĩ cần chỉ một giai đoạn bao gồm các hoạt động liên tiếp mà làm bệnh nhân cảm thấy vui vẻ nhất.

*Giá trị độ vui vẻ của một giai đoạn được tính bằng tích của giá trị cảm xúc nhỏ nhất nhân số tổng giá trị cảm xúc trong giai đoạn đó*.

Hãy giá trị độ vui vẻ nhất của một bệnh nhân được đưa ra khi tham gia các hoạt động.

## Dữ liệu vào: feelgood.inp

* Dòng đầu ghi số ***n*** là số hoạt động mà bệnh tham gia – (1 ≤ ***n*** ≤ 100 000).
* Dòng tiếp theo ghi ***n*** số thể hiện giá trị của bệnh nhân sau khi tham gia các hoạt động. Các giá trị này nằm trong khoảng từ 0 đến 106.

## Kết quả ra: feelgood.out

* Ghi một số duy nhất là giá trị độ vui vẻ nhất của một bệnh nhân trong một gia đoạn nào đó.

|  |  |  |
| --- | --- | --- |
| **feelgood.inp** | **feelgood.out** | **Chú thích** |
| 6  3 1 6 4 5 2 | 60 | Trong gia đoạn từ 3 đến 5 |

# Bài 9. TRỌNG SỐ KHOẢNG Tên file: WEIGHT.CPP

Định nghĩa trọng số của một dãy số nguyên là độ chênh lệch giữa phần tử lớn nhất và phần tử nhỏ nhất trong dãy.

Ví dụ trọng số của dãy (3,1,7,2) là 6, trọng số của dãy (40,40) là 0.

**Yêu cầu**: Cho dãy số nguyên 𝐴 = (𝑎*1*, 𝑎*2*, … , 𝑎𝑛). Hãy tính tổng trọng số của tất cả các dãy con gồm các phần tử liên tiếp trong 𝐴.

Ví dụ với 𝐴 = (1,2,3), những dãy con gồm các phần tử liên tiếp trong 𝐴 là:

* Dãy rỗng và các dãy (1), (2), (3): trọng số 0
* Dãy (1,2) và dãy (2,3): trọng số 1
* Dãy (1,2,3): trọng số 2
* Tổng trọng số cần tìm: 4

## INPUT: WEIGHT.INP

* Dòng 1 chứa số nguyên dương 𝑛 ≤ 106
* Dòng 2 chứa 𝑛 số nguyên dương 𝑎*1*, 𝑎*2*, … , 𝑎*𝑛* có giá trị không vượt quá 106.

*Các số trên một dòng của input file được ghi cách nhau ít nhất một dấu cách.*

## OUTPUT: WEIGHT.OUT

* Một số nguyên duy nhất là kết quả tìm tìm được

Ví dụ:

|  |  |
| --- | --- |
| **WEIGHT.INP** | **WEIGHT.OUT** |
| 4  3 1 7 2 | 31 |

# Bài 10. Quá tải Tên file: OVERLOAD.CPP

Trên đường cao tốc dẫn đến bến cảng có rất nhiều xe lưu thông. Các xe khi rời cảng được kiểm soát chặt không để xẩy ra hiện tượng chở quá tải, nhưng các xe chở hàng tới có thể vi phạm các quy định về tải trọng. Để phát hiện và bắt giữ các xe vi phạm người ta bố trí trên đường ***m*** thiết bị cân tự động, thiết bị thứ ***i*** đặt ở ki lô mét ***bi*** (***i*** = 1, 2, . . ., ***m***). Nếu xe có tải trọng lớn hơn mức được phép đi qua cân, các cảm biến sẽ được kích hoạt, các má phanh bật lên ôm sát bánh buộc xe phải dừng lại. Mỗi cân chỉ được kích hoạt tự động một lần, muốn tháo cân trả về trạng thái ban đầu phải có sự can thiệp trực tiếp của cán bộ vận hành.

Có ***n*** xe quá tải lưu thông trên đường, xe thứ ***j*** vào đường cao tốc ở km ***aj*** (***j*** = 1, 2, . . ., ***n***). Nếu một xe đang ở vị trí km ***x*** và vẫn đi được thì sau một đơn vị thời gian xe sẽ ở vị trí ***x***+1.

Với mỗi xe hãy xác định nó sẽ bị cân tự động nào phát hiện. Nếu xe không bị phát hiện thì đưa ra số -1.

## *Dữ liệu vào:* OVERLOAD.INP:

* Dòng đầu tiên chứa 2 số nguyên ***n*** và ***m*** (1 ≤ ***n***, ***m*** ≤ 105),
* Dòng thứ 2 chứa ***n*** số nguyên ***a1***, ***a2***, . . ., ***an*** (0 ≤ ***aj*** ≤ 109, ***j*** = 1, 2, . . ., ***n***)
* Dòng thứ 3 chứa ***m*** số nguyên ***b1***, ***b2***, . . ., ***bm*** (0 ≤ ***bi*** ≤ 109, ***i*** = 1, 2, . . ., ***m***)

## *Kết quả ra:* OVERLOAD.OUT

* ***n*** số nguyên, mỗi số trên một dòng, số thứ ***j*** xác định số thứ tự của cân phát hiện xe ***j***, ***j***=1, 2, . . ., ***n***.

***Ví dụ:***

|  |  |
| --- | --- |
| OVERLOAD.INP | OVERLOAD.OUT |
| 8 6  0 2 3 4 5 6 8 13  1 3 5 6 9 12 | 1  -1  2  6  3  4  5  -1 |

# Bài 11. Tích ba số tên file: Three.cpp

Cho dãy số và số nguyên .

Yêu cầu: Chọn ra 3 số có tích lớn nhất sao cho khoảng cách giữa vị trí của 2 số bất kỳ không vượt quá .

## Dữ liệu vào: THREE.INP

* Dòng đầu tiên chứa 2 số nguyên dương
* Dòng thứ 2 chứa số nguyên dương

## Dữ liệu ra: THREE.OUT

* Một giá trị là tích lớn nhất của 3 số tìm được.

Ví dụ:

|  |  |
| --- | --- |
| **THREE.INP** | **THREE.OUT** |
| 7 3  6 1 5 2 1 3 9 | 60 |

**Ràng buộc:**

* 20% số test có
* 30% số test khác có
* số test còn lại có

# Bài 12. Kéo xe Tên file: cardrag.cpp

Bãi để xe ô tô bị thu giữ vì vi phạm luật giao thông có dạng lưới ô vuông kích thước 𝑀×𝑁, mỗi xe nằm gọn trên một ô vuông theo một trong bốn hướng N, S, E, W (bắc, nam, đông, tây).

Vì một số nguyên nhân đặc biệt, quản lí bãi đỗ cần giải phóng nhanh chóng càng nhiều xe càng tốt. Muốn vậy cần đẩy các xe ra khỏi bãi mà không gây va chạm, tức là cần đẩy các xe theo hướng đỗ ban đầu đi ra khỏi bãi sao cho trong quá trình đẩy xe không đi vào ô vẫn còn xe. Một xe nếu đã bị di chuyển thì phải được kéo ra khỏi bãi.

Hãy xác định số lượng xe nhiều nhất có thể đẩy ra khỏi bãi với một trình tự đẩy xe tối ưu.

## Dữ liệu: cardrag.inp

Gồm nhiều tests, mỗi test cho trên nhóm dòng theo định dạng:

* Dòng 1: hai số nguyên 𝑀, 𝑁 (1 ≤ 𝑀, 𝑁 ≤ 2000);
* Dòng 2 … 𝑀 + 1: mỗi dòng ghi xâu độ dài 𝑁 chỉ gồm các kí tự trong tập {N S E W} thể hiện ô trống hay ô có xe đỗ theo hướng tương ứng.
* File dữ liệu kết thúc bằng dòng ghi hai số 0.

## Kết quả: cardrag.out

* Kết quả mỗi test ghi trên một dòng số nguyên là số lượng xe nhiều nhất có thể đẩy ra khỏi bãi.

Ví dụ

|  |  |
| --- | --- |
| cardrag.inp | cardrag.out |
| 3 4  .N.W  WWSS  EWEW  0 0 | 4 |