# **Bài 1. Bí kíp luyện rồng Tên file: dragon.\***

Sau khi không còn gì để đố 2 thằng em của mình nữa Tý quay ra chơi điện tử. Tý rất mê trò chơi điều khiển nhân vật Hiccup của mình chiến đấu cùng với những con rồng, vượt qua tất cả sẽ nhận được chứng chỉ dũng sĩ dragon. Hiccup chỉ có thể vượt qua 1 con rồng nếu như chỉ số chiến đấu của cậu lớn hơn chỉ số chiến đấu của con rồng đó, và sau khi giành chiến thắng chỉ số chiến đấu của cậu sẽ tăng thêm một mức nhất định tùy vào con rồng mà cậu đánh bại.

Tý muốn biết liệu Hiccup có thể nhận được chứng chỉ không? Bạn hãy giúp Tý với.

INPUT:

* Dòng đầu tiên gồm 2 số nguyên s và n, trong đó n là số con rồng Hiccup cần phải vượt qua (1 ≤ n ≤ 103) và s là chỉ số chiến đấu ban đầu của Hiccup (1 ≤ s ≤ 104).
* n dòng tiếp theo, dòng thứ i chứa cặp số (x[i], y[i]). Trong đó x[i] là chỉ số chiến đấu của con rồng thứ i (1 ≤ x[i] ≤ 104), y[i] là mức tăng chỉ số cho Hiccup nếu cậu đánh bại nó (0 ≤ y[i] ≤ 104).

OUTPUT:

* In ra “YES” nếu Hiccup nhận được chứng chỉ dũng sĩ dragon.
* Trường hợp ngược lại thì:
* Dòng 1 ghi ”NO” nếu Hiccup không nhận được chứng chỉ dũng sỹ dragon.
* Dòng 2 đưa ra số lượng rồng còn lại chưa bị tiêu diệt.

Ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
| Test 1 | | Test 2 | |
| INPUT | OUTPUT | INPUT | OUTPUT |
| 2 2  1 99  100 0 | YES | 10 1  100 100 | NO  1 |

# **Bài 2. Bài tập về nhà Tên file: homework.\***

Bạn An được thầy giáo giao cho N bài tập về nhà, mỗi bài tập có mức độ khó khác nhau được đánh giá bởi giá trị a[i].

Để làm được bài tập thứ i thì bạn An cần phải có kỹ năng lớn hơn hoặc bằng mức độ khó a[i]. Mỗi khi làm được bài độ khó là a[i] thì kỹ năng giải bài của cậu được tăng thêm một lượng là a[i]. Ban đầu kỹ năng làm bài của bạn An là S.

Hãy cho biết bạn An có thể làm được tối đa bao nhiêu bài tập.

INPUT: HOMEWORK.INP

* Dòng 1 chứa N và S (1 <= N <= 100, S <= 1000)
* Dòng 2 chứa N số nguyên a[1], a[2], ..., a[N]

Output: HOMEWORK.OUT

* Một số nguyên duy nhất là số lượng bài tập lớn nhất làm được.

Ví dụ

|  |  |
| --- | --- |
| HOMEWORK.INP | HOMEWORK.OUT |
| 5 2  18 2 5 14 3 | 3 |

# **Bài 3. Vắt sữa bò Tên file: milk2.\***

Vào một buổi sáng nông dân John sắp một đàn bò gồm ***n*** con bò để vắt sữa. Ông dự kiến là vào sáng hôm đó, con bò thứ ***i*** có khả năng sẽ vắt được ***ai*** lít sữa. Tuy nhiên đàn bò của ông có đặc điểm là cứ mỗi lần vắt sữa một con, những con còn lại trông thấy sợ quá nên sẽ bị giảm sản lượng mỗi con **01** lít sữa. Nếu vắt sữa con bò thứ nhất, ***n-1*** con còn lại bị giảm sản lượng. Sau đó vắt sữa con bò thứ hai thì ***n-2*** con còn lại bị giảm sản lượng....

Bạn hãy giúp nông dân John tính xem thứ tự vắt sữa bò như thế nào để số lượng sữa vắt được là nhiều nhất nhé.

Dữ liệu vào: gồm 2 dòng

- Dòng thứ nhất là số nguyên n (1 ≤ n ≤ 100) là số lượng con bò.

- Dòng thứ hai gồm n số nguyên a1, a2,..., an (1 ≤ ai ≤ 1000) là sản lượng sữa của các con bò.

Dữ liệu xuất:

- Là một số nguyên xác định số lít sữa nhiều nhất mà anh Bo có thể vắt được.

Ví dụ

|  |  |
| --- | --- |
| Milk2.INP | Milk2.OUT |
| 4 4 4 4 4 | 10 |
| 4 2 1 4 3 | 6 |

# **Bài 4. Xếp gạch Tên file: tile.\***

Nam có ***n*** viên gạch được đánh số từ ***1*** đến ***n***. Các viên gạch có độ cứng lần lượt là ***a1***, ***a2***,..., ***an***. Một viên gạch có độ cứng ***x*** nghĩa là Nam có thể chồng lên trên viên gạch đó tối đa ***x*** viên gạch khác, nếu chồng nhiều hơn thì viên gạch đó bị vỡ. Hỏi Nam có thể sắp được chồng gạch cao nhất là bao nhiêu?

Dữ liệu nhập:

- Dòng đầu tiên là số nguyên ***n*** (1 ≤ ***n*** ≤ 100) - là số viên gạch.

- Dòng tiếp theo gồm ***n*** số nguyên ***a1***, ***a2***,..., ***an*** (0 ≤ ***ai*** ≤ 100) mỗi số cách nhau một khoảng trắng.

Dữ liệu xuất

- Là số nguyên xác định chiều cao cao nhất của chồng gạch mà Nam sắp được.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 3  1 2 1 | 3 |
| 6  0 0 0 0 0 0 | 1 |

# **Bài 5. Thu mua sữa bò tên file: milk.\***

Một nhà máy chế biến sữa cần mua hàng ngày N lít sữa của M nông dân. Mỗi nông dân có thể cung cấp một lượng sữa và đưa ra giá bán sữa khác nhau.

Hãy tìm cách mua sữa sao cho đủ số lượng sữa cần cho hàng ngày với số tiền bỏ ra là ít nhất. Biết rằng tổng lượng sữa của các nông dân luôn đủ cung cấp cho nhà máy.

Dữ liệu vào:

* Dòng đầu tiên chứa hai số N (0 <= N <= 2.000.000) và M (0 <= M <= 5,000)
* M dòng tiếp theo, mỗi dòng chưa hai số nguyên Pi (0 <= Pi <= 1,000) và Ai (0 <= Ai <= 2,000,000), trong đó Pi là số tiền mà nông dân bán 1 lít, Ai là số lượng sữa mà nông dân thứ i có thể cung cấp

Kết quả:

* Một số nguyên duy nhất là số tiền ít nhất để mua sữa

Ví dụ

|  |  |  |
| --- | --- | --- |
| Milk.inp | Milk.out | Giải thích |
| 100 5  5 20  9 40  3 10  8 80  6 30 | 630 | Mua 20 lít sữa của nông dân 1 => 5\*20 = 100  Không mua sữa của nông dân 2  Mua 10 lít sữa của nông dân 3 => 3\*10 = 30  Mua 40 lít sữa của nông dân 4 => 8\*40 = 320  Mua 30 lít sữa của nông dân 5 => 6\*30 = 180  Tổng số tiền mua = 100 + 30 + 320 + 180 = 630 |

# **Bài 6. Tìm số Tên file: find.\***

Tìm số nguyên dương K nhỏ nhất sao cho tích các chữ số của K bằng số nguyên M cho trước.

Nếu không tìm được K, in ra -1.

Input

- Một số nguyên M (0 ≤ M ≤ 109).

Output

- Số nguyên dương K là kết quả của bài toán.

- Nếu không tìm được K, in ra -1.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 0 | 10 |
| 12 | 26 |

# **Bài 7. Cắm điện Tên file: election.\***

Trong nhà Nam hiện đang có ***n*** ổ cắm điện rời. Số lượng chỗ cắm trên mỗi ổ cắm điện này lần lượt là ***a1, a2, a3,…, an***  chỗ cắm. Trên tường nhà Nam có một chỗ cắm cố định đang có điện. Vậy để cho một ổ cắm điện rời có điện thì phải cắm ổ cắm đó vào chỗ cắm cố định trên tường. Chúng ta cũng có thể cắm ổ cắm điện rời này vào một ổ cắm điện rời khác đang có điện.

Nam có ***m*** thiết bị sử dụng điện, để sử dụng thì các thiết bị này cần được cắm vào ổ cắm trên tường hoặc ổ cắm rời đang có điện.

Bạn hãy giúp Nam tìm ra số ổ cắm rời ít nhất cần dùng để có thể sử dụng tất cả ***m*** thiết bị điện này.

Dữ liệu vào:

- Dòng thứ nhất gồm 2 số nguyên ***n, m*** cách nhau một khoảng trắng, dữ liệu vào đảm bảo 1 ≤ ***n, m*** ≤ 50, ***n*** là số lượng ổ cắm và ***m*** là số lượng thiết bị.

- Dòng thứ hai gồm n số nguyên ***a1, a2, …, an*** là số chỗ cắm trên các ổ cắm rời tương ứng, mỗi số cách nhau một khoảng trắng, dữ liệu vào đảm bảo 1 ≤ ***ai*** ≤ 50.

Dữ liệu ra:

- Là số nguyên cho biết số ổ cắm rời ít nhất cần sử dụng là bao nhiêu. Nếu đã sử dụng hết tất cả ổ cắm rời mà vẫn không đủ, in ra -1.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 4 7  3 3 2 4 | 3 |
| 5 5  1 3 1 2 1 | -1 |

# **Bài 8. Taxi Tên file: taxi.\***

Sau buổi học, có ***n*** nhóm bạn rủ nhau đến chúc mừng sinh nhật An. Nhóm bạn thứ ***i*** có ***ai*** người (1 ≤ ***ai*** ≤ 4). Họ quyết định gọi những chiếc xe taxi để đi. Những chiếc taxi chỉ chở tối đa được 4 khách mà thôi.

Hãy cho biết các nhóm bạn này cần gọi tối thiểu bao nhiêu chiếc taxi, biết rằng những người trong cùng một nhóm luôn đi cùng trên một chiếc taxi, và một chiếc taxi có thể chở nhiều nhóm khác nhau.

Input:

- Dòng 1: chứa số nguyên dương ***n*** (1 ≤ ***n*** ≤ 105)

- Dòng 2: chứa ***n*** số nguyên dương ***a1, a2, … an*** – là số lượng người của mỗi nhóm bạn.

Output

- Một số nguyên dương duy nhất là số lượng taxi.

Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 5  1 2 4 3 3 | 4 |
| 8  2 3 4 4 2 1 3 1 | 5 |

# **Bài 9. CÁC ĐỒNG XU Tên chương trình: COINS.\***

Steve ngồi bên cửa sổ rất lâu quan sát lũ quạ. Chúng là loài chim thông minh, rất thích các vật lóng lánh và hay tha những thứ này về tổ. Hôm nay, không biết kiếm được ở đâu, chúng tha về các đồng xu. Có tất cả ***n*** con quạ. Con quạ thứ ***i*** đã mang về ***ai*** đồng xu. Nhìn vào vị trí của tổ trên cành Steve hiểu rằng nếu có ***bi*** đồng xu thì tổ của quạ thứ ***i*** sẽ bị lật, rơi xuống đất và Steve sẽ nhặt được hết xu trong tổ. Steve đang có trong túi ***m*** đồng xu và có tài ném đâu trúng đó. Bây giờ chính là lúc cái tài lẻ này phát huy tác dụng.

***Yêu cầu:*** Hãy xác định số tiền tối đa mà Steve sẽ có được.

***Dữ liệu:*** Vào từ file văn bản COINS.INP:

* Dòng đầu tiên chứa 2 số nguyên ***n*** và ***m*** (1 ≤ ***n*** ≤ 1 000, 0 ≤ ***m*** ≤ 1 000),
* Dòng thứ 2 chứa ***n*** số ***a1***, ***a2***, . . ., ***an*** (0 ≤ ***ai*** ≤ 1 000, ***i*** = 1 ÷ ***n***),
* Dòng thứ 3 chứa ***n*** số ***b1***, ***b2***, . . ., ***bn*** (***ai***<***bi*** ≤ 1 000, ***i*** = 1 ÷ ***n***).

***Kết quả:*** Đưa ra file văn bản COINS.OUT một số nguyên – số tiền tối đa mà Steve sẽ có.

***Ví dụ:***

|  |  |  |
| --- | --- | --- |
| COINS.INP |  | COINS.OUT |
| **2 3**  **1 2**  **4 6** |  | **6** |

# **Bài 10. NỐI ĐIỂM ĐEN TRẮNG Tên file: bwpoints.\***

Trên trục số thực cho n điểm đen và n điểm trắng hoàn toàn phân biệt. Các điểm đen có tọa độ nguyên *a1, a2, …, an* còn các điểm trắng có tọa độ nguyên *b1, b2, …, bn*. Người ta muốn chọn ra k điểm đen và k điểm trắng để nối mỗi một điểm đen với một điểm trắng sao cho *k* đoạn thẳng tạo được đôi một không có điểm chung.

**Yêu cầu:** Cho tọa độ của n điểm đen *a1, a2, …, an* và tọa độ của điểm trắng *b1, b2, …, bn*. Hãy tìm giá trị k lớn nhất thỏa mãn yêu cầu trên.

**Dữ liệu: bwpoints.inp**

* Dòng thứ nhất chứa số nguyên dương n (n <= 105).
* Dòng thứ hai chứa các số *a1, a2, …, an* (|ai| <= 109, i = 1, 2,…, n)
* Dòng thứ ba chứa các số *b1, b2, …, bn* (|bi| <= 109, i = 1, 2,…, n)

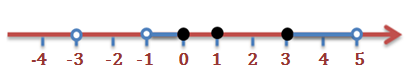
      Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

**Kết quả:** **bwpoints.out**

* Ghi ra một số nguyên duy nhất là số *k* lớn nhất tìm được

**Ví dụ:**

|  |  |
| --- | --- |
| **bwpoints.inp** | bwpoints.out |
| 3  0 3 1  -3 5 -1 | 2 |



**Ràng buộc:** 50% số test ứng với 50% số điểm của bài có 1 <= n <= 100

# **Bài 11. Những người lái xe buýt Tên file: BUSDRIVER.\***

Trong một thành phố có ***n*** tài xế xe buýt. Ngoài ra còn có ***n*** tuyến xe buýt ban ngày và các tuyến xe buýt ban tối với chiều dài khác nhau. Mỗi tài xế được chỉ định một tuyến đường ban ngày và một tuyến đường ban tối. Đối với bất kỳ tài xế nào, nếu tổng chiều dài tuyến của anh ta vượt quá ***d*** giờ, anh ta phải được trả lương làm thêm giờ cho mỗi giờ sau ***d*** giờ đầu tiên với một giờ ***r*** đồng / giờ.

Nhiệm vụ của bạn là chỉ định một tuyến đường ban ngày và một tuyến đường ban tối cho mỗi tài xế xe buýt để tổng số tiền làm thêm giờ mà công ty phải trả cho họ càng nhỏ càng tốt.

**INPUT: BUSDRIVER.INP**

* Gồm nhiều bộ test, mỗi bộ test có dạng như sau:
  + Dòng đầu tiên chứa 3 số nguyên ***n***, ***d***, ***r*** – được mô tả như trên
  + Dòng thứ hai chứa ***n*** số nguyên (có giá trị không lớn hơn 10000) thể hiện độ dài của ***n*** tuyến xe buýt ban ngày.
  + Dòng thứ ba chứa ***n*** số nguyên (có giá trị không lớn hơn 10000) thể hiện độ dài của ***n*** tuyến xe buýt ban tối.
* Kết thúc input là 3 số 0.

**OUTPUT: BUSDRIVER.OUT**

* Gồm nhiều dòng, mỗi dòng là số tiền phải trả nhỏ nhất cho các lái xe buyt tương ứng với các test trong input.
* Giới hạn:
  + 1 ≤ ***n*** ≤ 100
  + 1 ≤ ***d*** ≤ 10000
  + 1 ≤ ***r*** ≤ 5

***Ví dụ:***

|  |  |
| --- | --- |
| **BUSDRIVER.INP** | **BUSDRIVER.OUT** |
| 2 20 5  10 15  10 15  2 20 5  10 10  10 10  3 4 2  4 2 3  1 3 2  0 0 0 | 50  0  6 |

# **Bài 12. Bảng chữ cái Tên file: ALPHABET.\***

Trong bài toán này, bạn phải vẽ một hình vuông bằng cách sử dụng bảng chữ cái tiếng Anh viết hoa.

Để chính xác hơn, bạn sẽ được cung cấp một hình vuông với một số khối trống và các khối khác đã được điền sẵn cho bạn với một số chữ cái để làm cho công việc của bạn dễ dàng hơn. Bạn phải chèn ký tự vào mọi ô trống để toàn bộ lưới được lấp đầy bằng bảng chữ cái. Khi làm như vậy bạn phải đáp ứng các quy tắc sau:

1. Đảm bảo không có ô liền kề nào chứa cùng một chữ cái; hai ô liền kề nếu chúng có chung cạnh.

2. Có thể có nhiều cách để lấp đầy lưới điện. Bạn phải đảm bảo bạn điền vào bảng sao cho các chữ cái có thứ tự từ điển nhỏ nhất. Ở đây, hai lưới được kiểm tra theo thứ tự hàng khi so sánh theo từ điển.

INPUT: ALPHABET.INP

* Dòng đầu tiên chứa số nguyên dương T, số lượng bộ test
* Mỗi bộ test có dạng:
  + Dòng đầu chứa số nguyên dương n (n ≤ 20), kích thước của bảng hình vuông.
  + n dòng tiếp theo, mỗi dòng chứa n ký tự ‘.’ (thể hiện ô trống) hoặc ký tự nào đó thuộc đoạn [‘A’, ‘Z’] (thể hiện ô đã được điền sẵn).

OUTPUT: ALPHABET.OUT

* Tương ứng mỗi test, tại test thứ i ghi ra từ “Case i:’
* Các dòng tiếp theo ghi ra bảng chữ cái thoản mãn yêu cầu đề bài

Ví dụ:

|  |  |
| --- | --- |
| ALPHABET.INP | ALPHABET.OUT |
| 2  3  ...  ...  ...  3  ...  A..  ... | Case 1:  ABA  BAB  ABA  Case 2:  BAB  ABA  BAB |

# **Bài 13. Sơn bảng Tên file: PAINT.CPP**

Một bảng số kích thước ***m*** x ***n***, ban đầu bảng số được phủ toàn bộ là màu trắng. Người ta tiến hành sơn bảng số này như sau:

* Chọn một bảng con kích thước ***r*** x ***c*** bất kỳ*.*
* Sơn toàn bộ bảng con vừa chọn, màu trắng thành màu đen và màu đen thành màu trắng.
* Lặp lại các bước trên cho đến khi được một mẫu cho trước.

Bạn hãy cho biết cần tối thiểu bao nhiêu lần sơn để được mẫu cho trước.

**INPUT: PAINT.INP**

* Gồm nhiều bộ test, mỗi bộ test bao gồm:
  + Dòng 1 chứa 4 số nguyên dương ***m, n, r, c*** (1 ≤ ***r*** ≤ ***n*** ≤ 100, 1 ≤ ***c*** ≤ ***m*** ≤ 100).
  + Các dòng tiếp theo thể hiện bảng mẫu. Bản mẫu gồm ***m*** dòng, mỗi dòng chứa ***n*** ký tự 0 hoặc 1, trong đó 0 là màu trắng và 1 là màu đen.
* Kết thúc bộ test là 4 số 0 0 0 0.

**OUTPUT: PAINT.OUT**

* Đưa ra số lượng sơn bảng ít nhất có thể, trong trường hợp không tạo được mẫu cho trước thì đưa ra -1.

***Ví dụ:***

|  |  |
| --- | --- |
| PAINT.INP | PAINT.OUT |
| 3 3 1 1  010  101  010  4 3 2 1  011  110  011  110  3 4 2 2  0110  0111  0000  0 0 0 0 | 4  6  -1 |