**DSA03001.**

Tại ngân hàng có các mệnh giá bằng 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000. Tổng số tiền cần đổi có giá trị bằng N.  Hãy xác định xem có ít nhất bao nhiêu tờ tiền sau khi đổi tiền?

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 50).  Mỗi test gồm 1 số nguyên N ( 1 ≤ N ≤ 100 000).

**Output:**Với mỗi test, in ra đáp án trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 2  70  121 | 2  3 |

**DSA03002.**

Trong một buổi học toán, giáo viên viết 2 số nguyên, A và B, và yêu cầu Tèo thực hiện phép cộng. Tèo không bao giờ tính toán sai, nhưng thỉnh thoảng cậu ta chép các con số một cách không chính xác. Lỗi duy nhất của là ghi nhầm '5' thành '6' hoặc ngược lại.  Cho hai số, A và B, tính tổng nhỏ nhất và lớn nhất mà Tèo có thể nhận được.

**Input:** Có một dòng chứa hai số nguyên dương A và B ( 1 ≤ A, B ≤ 1 000 000).

**Output:** In ra 2 số nguyên cách nhau một dấu cách, tổng nhỏ nhất và lớn nhất có thể nhận được.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Test 1** | **Test 2** | **Test 3** |
| Input: 11 25  Ouput: 36 37 | Input: 1430 4862  Ouput: 6282 6292 | Input: 16796 58786  Ouput: 74580 85582 |

**DSA03003.**

Cho mảng A[] gồm N phần tử.Nhiệm vụ của bạn là tìm C:\Users\Xuan Loc\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\122A0073.tmp bằng cách sắp đặt lại các phần tử trong mảng. Chú ý, kết quả của bài toán có thể rất lớn vì vậy bạn hãy đưa ra kết quả lấy modulo với 109+7.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N, A[i] ≤107.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 5 5 3 2 4 1 3 1 2 3 | 40 8 |

**DSA03004**

Cho mảng A[] gồm các số từ 0 đến 9. Nhiệm vụ của bạn là tìm tổng nhỏ nhất của hai số được tạo bởi các số trong mảng A[]. Chú ý, tất cả các số trong mảng A[] đều được sử dụng để tạo nên hai số.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N ≤20; 0≤A[i]≤9.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví  dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 6 6 8 4 5 2 3 5 5 3 0 7 4 | 604 82 |

**DSA03005.**

Cho mảng A[] gồm N số nguyên không âm và số K. Nhiệm vụ của bạn là hãy chia mảng A[] thành hai mảng con có kích cỡ K và N-K sao cho hiệu giữa tổng hai mảng con là lớn nhất. Ví dụ với mảng A[] = {8, 4, 5, 2, 10}, K=2 ta có kết quả là 17 vì mảng A[] được chia thành hai mảng {4, 2} và { 8, 5,10} có hiệu của hai mảng con là 23-6=17 là lớn nhất.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N và số K; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, K, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ K<N ≤50; 0≤A[i]≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 5 2 8 4 5 2 10 8 3 1 1 1 1 1 1 1 1 | 17 2 |

**DSA03006**.

Cho mảng A[] gồm N số và thực hiện các thao tác theo nguyên tắc dưới đây:

* Ta chọn một mảng con sao cho phần tử ở giữa của mảng con cũng là phần tử ở giữa của mảng A[] (trong trường hợp N lẻ).
* Đảo ngược mảng con đã chọn trong mảng A[]. Ta được phép chọn mảng con và phép đảo ngược mảng con bao nhiêu lần tùy ý.

Ví dụ với mảng A[] = {1, 6, **3, 4, 5**, 2, 7} ta có câu trả lời là Yes vì: ta chọn mảng con {3, 4, 5} và đảo ngược để nhận được mảng A[]={1, 6, **5, 4, 3**, 2, 7}, chọn tiếp mảng con {6, 5, 4, 3, 2} và đảo ngược ta nhận được mảng A[]={1, 2, 3, 4, 5, 6, 7}. Hãy cho biết ta có thể sắp xếp được mảng A[] bằng cách thực hiện các thao tác kể trên hay không?

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i] tương ứng với các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N ≤50; 0≤A[i]≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 7 1 6 3 4 5 2 7 7 1 6 3 4 5 7 2 | Yes No |

**DSA03007**

Cho mảng A[], B[] đều có N phần tử. Nhiệm vụ của bạn là tìm giá trị nhỏ nhất của  biểu thức P = A[0]\*B[0] + A[1]\*B[1] + ..+A[N-1]\*B[N-1] bằng cách tráo đổi vị trí các phần tử của cả mảng A[] và B[].

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo đưa vào N số A[i]; dòng cuối cùng đưa vào N số B[i] các số được viết cách nhau một vài khoảng trống.
* T, N, A[i], B[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤ N ≤107; 0≤A[i], B[i] ≤1018.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 7 1 6 3 4 5 2 7  1 1 1 2 3 4 3 7 1 6 3 5 5 2 2  0 1 9 0 1 2 3 | 45  27 |

**DSA03008**

Cho hệ gồm N hành động. Mỗi hành động được biểu diễn như một bộ đôi <Si, Fi> tương ứng với thời gian bắt đầu và thời gian kết thúc của mỗi hành động. Hãy tìm phương án thực hiện nhiều nhất các hành động được thực hiện bởi một máy hoặc một người sao cho hệ không xảy ra mâu thuẫn.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng thứ nhất đưa vào số lượng hành động N; dòng tiếp theo đưa vào N số Si tương ứng với thời gian bắt đầu mỗi hành động; dòng cuối cùng đưa vào N số Fi tương ứng với thời gian kết thúc mỗi hành động; các số được viết cách nhau một vài khoảng trống.
* T, N, Si, Fi thỏa mãn ràng buộc: 1≤T≤100;  1≤N, Fi, Si≤1000.

**Output:**

* Đưa số lượng lớn nhất các hành động có thể được thực thi bởi một máy hoặc một người.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 1  6  1  3  0  5  8  5  2  4  6  7  9  9 | 4 |

**DSA03009**

Cho N công việc. Mỗi công việc được biểu diễn như một bộ 3 số nguyên dương <JobId, Deadline, Profit>, trong đó JobId là mã của việc, Deadline là thời gian kết thúc của việc, Profit là lợi nhuận đem lại nếu hoàn thành việc đó đúng thời gian. Thời gian để hoàn toàn mỗi công việc là **1 đơn vị thời gian**. Hãy cho biết lợi nhuận lớn nhất có thể thực hiện các việc với giả thiết mỗi việc được thực hiện đơn lẻ.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là số lượng Job N; phần thứ hai đưa vào 3×N số tương ứng với N job.
* T, N, JobId, Deadline, Profit thỏa mãn ràng buộc:1≤T≤100;  1≤N≤1000; 1≤ JobId ≤1000; 1≤ Deadline ≤1000; 1≤ Profit ≤1000.

**Output:**

* Đưa số lượng công việc tương ứng và lợi nhuận lớn nhất có thể đạt được.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 4 1 4 20  2 1 10  3 1 40  4 1 30 5 1 2 100  2 1 19  3 2 27  4 1 25  5 1 15 | 2 60 2 127 |

**DSA03010.**

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j].

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào số lượng sợi dây N; dòng tiếp theo đưa vào N số A[i] là độ dài của các sợi dây; các số được viết cách nhau một vài khoảng trống.
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤106; 0≤A[i]≤106.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 4 4 3 2 6 5 4 2 7 6 9 | 29 62 |

**DSA03011**

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j].

**Dữ liệu vào**

Dòng đầu ghi số bộ test T (T<10). Mỗi bộ test gồm 2 dòng. Dòng đầu tiên là số nguyên N (N ≤ 2\*106).

Dòng tiếp theo gồm N số nguyên dương c[i] ( 1 ≤ A[i] ≤ 109).

**Kết quả**

In ra đáp án của bộ test trên từng dòng, theo modulo 109+7.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 7  2 4 1 2 10 2 3 | 59 |

**DSA03012.**

Cho xâu ký tự S bao gồm các ký tự in thường. Nhiệm vụ của bạn là kiểm tra xem ta có thể sắp đặt lại các ký tự trong S để hai ký tự giống nhau đều không kề nhau hay không? Đưa ra 1 nếu có thể sắp đặt lại các ký tự trong S thỏa mãn yêu cầu bài toán, ngược lại đưa ra -1.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
* T, S thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 3 geeksforgeeks bbbabaaacd bbbbb | 1 1  -1 |

**DSA03013.**

Cho xâu ký tự S bao gồm các ký tự in thường và số D. Nhiệm vụ của bạn là kiểm tra xem ta có thể sắp đặt lại các ký tự trong S để tất cả các ký tự giống nhau đều có khoảng cách là D hay không? Đưa ra 1 nếu có thể sắp đặt lại các ký tự trong S thỏa mãn yêu cầu bài toán, ngược lại đưa ra -1.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số D; dòng tiếp theo là xâu S.
* T, S, D thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000; 1≤D≤100.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 2  ABB 2 AAA | 1 -1 |

**DSA03014.**

Một số X được gọi là số khối lập phương nếu X là lũy thừa bậc 3 của số Y (X= Y3). Cho số nguyên dương N, nhiệm vụ của bạn là tìm số khối lập phương lớn nhất bằng cách loại bỏ đi các chữ số của N. Ví dụ số 4125 ta có kết quả là 125 = 53.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số tự nhiên N được viết trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100;  1≤N≤1018.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng. Nếu không tìm được đáp án in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 4125 976 | 125 -1 |

**DSA03015**

Giả sử bạn là một người nghèo trong địa phương của bạn. Địa phương của bạn có duy nhất một cửa hàng bán lương thực. Cửa hàng của bạn mở cửa tất cả các ngày trong tuần ngoại trừ chủ nhật. Cho bộ ba số N, S, M thỏa mãn ràng buộc sau:

* N : số đơn vị lương thực nhiều nhất bạn có thể mua trong ngày.
* S : số lượng ngày bạn cần được sử dụng lương thực để tồn tại.
* M : số đơn vị lương thực cần có mỗi ngày để bạn tồn tại.

Giả sử bạn đang ở ngày thứ 2 trong tuần và cần tồn tại trong S ngày tới. Hãy cho biết số lượng ngày ít nhất bạn cần phải mua lương thực từ của hàng để tồn tại hoặc bạn sẽ bị chết đói trong S ngày tới.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ 3 số N, S, M được viết trên một dòng.
* T, N, S, M thỏa mãn ràng buộc: 1≤T≤100;  1≤N, S, M ≤30.

**Output:**

* Đưa ra số ngày ít nhất bạn có thể mua lương thực để tồn tại hoặc đưa ra -1 nếu bạn bị chết đói.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 16 10 2 20 10 30 | 2 -1 |

**DSA03016.**

Cho hai số nguyên dương S và D, trong đó S là tổng các chữ số và D là số các chữ số của một số. Nhiệm vụ của bạn là tìm số nhỏ nhất thỏa mãn S và D? Ví dụ với S = 9, D = 2 ta có số nhỏ nhất thỏa mãn S và D là 18.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ 2 số S và D được viết trên một dòng.
* T, S, D thỏa mãn ràng buộc: 1≤T≤100;  1≤ S,D≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng. Nếu không có đáp án, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 9 2 20 3 | 18 299 |

**DSA03017.**

Cho xâu ký tự S. Ta gọi giá trị của xâu S là tổng bình phương số lần xuất hiện mỗi ký tự trong S. Hãy tìm giá trị nhỏ nhất của xâu S sau khi thực hiện K lần loại bỏ ký tự.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là số K; phần thứ hai là một xâu ký tự S được viết trên một dòng.
* T, S, K thỏa mãn ràng buộc: 1≤T≤100;  1≤length(S)≤10000; 1≤K≤1000.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2 2  ABCCBC 2  AAAB | 6 2 |

**DSA03018.**

Hoàng yêu thích các số may mắn. Ta biết rằng một số là *số may mắn* nếu biểu diễn thập phân của nó chỉ chứa các chữ số may mắn là 4 và 7. Ví dụ, các số 47, 744, 4 là số may mắn và 5, 17, 467 không phải. Hoàng muốn tìm số may mắn bé nhất có tổng các chữ số bằng n. Hãy giúp anh ấy

**Dữ liệu vào:**Dòng đầu ghi số bộ test, mỗi bộ test có một dòng chứa số nguyên *n* (1 ≤ *n* ≤ 106) — tổng các chữ số của số may mắn cần tìm.

**Kết quả:**In ra trên 1 dòng số may mắn bé nhất, mà tổng các chữ số bằng n. Nếu không tồn tại số thỏa mãn, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  11  10 | 47  -1 |

**DSA03019.**

Một phân số đơn vị nếu tử số của phân số đó là 1. Mọi phân số nguyên dương đều có thể biểu diễn thành tổng các phân số đơn vị. Ví dụ 2/3 = 1/2  + 1/6. Cho phân số nguyên dương P/Q bất kỳ (P < Q), hãy biểu diễn phân số nguyên dương thành tổng phân số đơn vị với số hạng tử là ít nhất.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ đôi tử số P và mẫu số Q của phân số nguyên dương được viết trên một dòng.
* T, P, Q thỏa mãn ràng buộc: 1≤T≤100;  1≤P, Q≤100.

**Output:**

* Đưa ra đáp án tìm được trên 1 dòng, theo dạng “1/a + 1/b + …”

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  2 3  1 3 | 1/2 + 1/6  1/3 |

**DSA03020.**

Cho một mảng S gồm 2×N ký tự, trong đó có N ký tự ‘[’ và N ký tự ‘]’. Xâu S được gọi là viết đúng nếu S có dạng S2[S1] trong đó S, S2 là các xâu viết đúng. Nhiệm vụ của bạn là tìm số các phép đổi chỗ ít nhất các ký tự kề nhau của xâu S viết sai để  S trở thành viết đúng. Ví dụ với xâu S =”[]][][” ta có số phép đổi chỗ kề nhau ít nhất là 2.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một xâu S viết sai theo nguyên tắc kể trên.
* T, S thòa mãn ràng buộc: 1≤T≤100; 1≤length(S)≤100000.

**Output:**

* Đưa kết quả trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 2  []][][  [][][] | 2 0 |

**DSA03022**.

Cho dãy số A gồm N phần tử là các số nguyên. Hãy tính tích lớn nhất của **2 hoặc 3** phần tử trong dãy.

**Input**

Dòng đầu tiên ghi số N (3 ≤ N ≤ 10000)

Dòng thứ 2 ghi N số của dãy A (|Ai| ≤ 1000)

**Outpput**

Ghi ra kết quả trên một dòng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  5 10 -2 3 5 2 | 250 |