Cho ma trận vuông A cỡ N\*N. Một ma trận vuông B nhỏ hơn cỡ M\*M có thể dùng làm “cửa sổ trượt” trên ma trận A nếu M là ước số của N.

Hãy thực hiện tính tích chập của ma trận B với từng “khung cửa số” tương ứng trên ma trận A. Tích chập được hiểu là tính giá trị tích từng vị trí tương ứng trên 2 ma trận kích thước bằng nhau.

Xem ví dụ để hiểu rõ hơn.

**Input**

Dòng đầu ghi số N (3 < N < 100) . Tiếp theo là N dòng ghi ma trận A. Các giá trị đều nguyên dương và không quá 1000.

Tiếp theo là một dòng ghi số M (1 < M <10). Tiếp theo là M dòng ghi ma trận B. Các giá trị lớn hơn hoặc bằng 0 và không quá 20.

Dữ liệu vào đảm bảo M là ước số của N.

**Output**

Ghi ra N dòng mô tả ma trận kết quả.

|  |  |
| --- | --- |
| 4  1 2 3 4  5 6 7 8  9 10 11 12  13 14 15 16  2  1 0  0 2 | 1 0 3 0  0 12 0 16  9 0 11 0  0 28 0 32 |

Cho ma trận vuông A cỡ N\*N. Một ma trận vuông B nhỏ hơn cỡ M\*M có thể dùng làm “cửa sổ trượt” trên ma trận A nếu M là ước số của N.

Hãy thực hiện tính tích chập của ma trận B với từng “khung cửa số” tương ứng trên ma trận A. Tích chập được hiểu là tính giá trị tích từng vị trí tương ứng trên 2 ma trận kích thước bằng nhau.

Xem ví dụ để hiểu rõ hơn.

**Input**

Dòng đầu ghi số N (3 < N < 100) . Tiếp theo là N dòng ghi ma trận A. Các giá trị đều nguyên dương và không quá 1000.

Tiếp theo là một dòng ghi số M (1 < M <10). Tiếp theo là M dòng ghi ma trận B. Các giá trị lớn hơn hoặc bằng 0 và không quá 20.

Dữ liệu vào đảm bảo M là ước số của N.

**Output**

Ghi ra N dòng mô tả ma trận kết quả.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  1 2 3 4  5 6 7 8  9 10 11 12  13 14 15 16  2  1 0  0 2 | 1 0 3 0  0 12 0 16  9 0 11 0  0 28 0 |

Cơ số từ 2 đến 36 được xây dựng từ 10 chữ số (0 đến 9) và 26 chữ cái Tiếng Anh in hoa (‘A’ đến ‘Z’).

Hãy viết chương trình chuyển một số nguyên dương N trong cơ số 10 sang cơ số b. Trong đó N không quá 100.000, 2 ≤ b ≤ 36.

**Input**

Dòng đầu ghi số bộ test, không quá 10.

Mỗi bộ test ghi 2 số N và b.

Nlà một số nguyên dương N trong cơ số 10, không quá 100.000.  2 ≤ b ≤ 36

**Output**

Với mỗi bộ test ghi ra kết quả đổi cơ số tương ứng.

**Ví dụ**

|  |  |
| --- | --- |
|  |  |
| 3  10 2  2021 2  31 16 | 1010  11111100101  1F |

Cho một bảng cỡ N x N, mỗi ô ghi giá trị số 0 hoặc 1. Chỉ số hàng và cột đánh số từ 1 đến N. Phép đảo giá trị (swap) trên một phần của bảng được được định nghĩa là đảo ngược giá trị tất cả các ô trong phần đó, 0 biến đổi thành 1, và 1 biến đổi thành 0.

Giả sử mỗi bước, bạn được phép swap tất cả các số của một hình chữ nhật con có điểm trái trên là (1, 1).

Hãy tính xem ít nhất bao nhiêu bước để có thể đưa bảng về trạng thái chứa toàn số 0.

**Input:**

Dòng đầu tiên là số nguyên N (1 ≤ N ≤ 10).

N dòng tiếp theo, mỗi dòng gồm N kí tự mô tả một hàng của bảng.

**Output:**

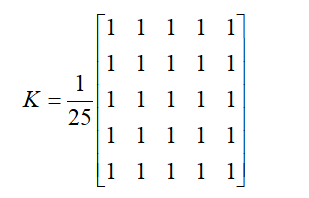
In ra đáp án tìm được.

3 2  
001

111

111

Phương pháp làm mịn ảnh được thực hiện bằng cách sử dụng phép tích chập (convolution) giữa ma trận ảnh và một ma trận kernel có dạng:



Ma trận kernel trong ví dụ trên có kích thước bằng 5. Với ma trận kernel có kích thước L = 2k + 1, giá trị điểm ảnh (i,j) của ma trận mới sẽ bằng tổng của (2k + 1) x (2k + 1) phần tử (i+u, j+v) với mọi –k ≤ u,v ≤ k, sau đó chia cho (2k + 1) x (2k + 1). Kết quả điểm ảnh mới thu được sau khi thực hiện phép chia sẽ được làm tròn xuống.

Cho ma trận ảnh đầu vào và kích thước L của ma trận kernel, nhiệm vụ của bạn là hãy in ra ma trận ảnh sau khi được làm mịn.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 10).

Mỗi test bắt đầu bởi hai số nguyên N, M và L (3 ≤ N,M ≤ 500; L ≤ min(n,m)). L được đảm bảo là một số nguyên lẻ.

Kế tiếp là N dòng, mỗi dòng gồm M số nguyên mô tả ma trận ảnh đầu vào, có giá trị trong phạm vi từ 0 tới 255.

**Output:**

Với mỗi test, hãy in ra ma trận ảnh sau khi đã được làm mịn.

**Ví dụ:**

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

Giải thích test 1: Giá trị phần tử (1,1) = floor[(2+1+0+3+2+1+4+5+2) / 9] = floor [20/9] = 2.

Cho dữ liệu vào dạng văn bản, với không quá 1000 số nguyên dương, các số không quá 50 chữ số. Hãy liệt kê các số thuận nghịch khác nhau theo thứ tự giảm dần và số lần xuất hiện của nó.

**Input**

Luồng vào dạng văn bản với không quá 1000 số nguyên dương. Không biết trước số dòng hoặc số lượng chính xác bao nhiêu số nguyên.

**Output**

Ghi ra các số thuận nghịch theo thứ tự giá trị giảm dần và số lần xuất hiện của nó, mỗi số trên một dòng. Không tính các số có 1 chữ số.

Nam viết bảng chữ cái 2 lần lên trên một vòng tròn, mỗi kí tự xuất hiện đúng 2 lần. Sau đó nối lần lượt các kí tự giống nhau lại. Tổng cộng có 26 đoạn thẳng.

Hình vẽ quá chằng chịt, Nam muốn đố các bạn xem có tất cả bao nhiêu giao điểm?

Một giao điểm được tính khi hai đường thẳng của một cặp kí tự cắt nhau.

**Input**

Gồm một xâu có đúng 52 kí tự in hoa. Mỗi kí tự xuất hiện đúng 2 lần.

**Output**

In ra đáp án tìm được.

|  |  |
| --- | --- |
| ABCCABDDEEFFGGHHIIJJKKLLMMNNOOPPQQRRSSTTUUVVWWXXYYZZ | 1 |

Giải thích test: Chỉ có duy nhất cặp kí tự ‘A’, ‘B’ tạo ra 2 đoạn thẳng cắt nhau.

Một thời không quá xa, điện thoại di động với chỉ các tính năng nghe, gọi, nhắn tin vẫn còn chiếm đại đa số thiết bị cầm tay tại Việt Nam. Khi nhắn tin, người nhắn sẽ bấm các phím số một đến bốn lần liên tiếp tương ứng với ký tự đi kèm ghi trên đó.

Cụ thể: các số và chữ cái tương ứng gồm:

2: ABC, 3: DEF, 4: GHI, 5: JKL

6: MNO, 7: PQRS, 8: TUV, 9: WXYZ

Cho trước dãy ký tự mô tả tin nhắn (không tính các ký tự khác ngoài danh sách nêu trên). Hãy kiểm tra xem dãy số được nhấn ứng với dãy ký tự đó có phải số thuận nghịch hay không (chỉ xét tương ứng giữa số và ký tự, không tính số đó được nhấn bao nhiêu lần, ví dụ tất cả A,B,C,a,b,c đều chỉ là một chữ số 2).

**Input**

Dòng đầu tiên là số bộ test, không quá 1000.

Mỗi test là dãy ký tự mô tả tin nhắn.

**Output**

Ghi ra kết quả kiểm tra, YES nếu dãy số là thuận nghịch, NO nếu ngược lại.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  BOHIMA  IamACoder | YES  NO |

|  |  |
| --- | --- |
| 12321 456 12321 34 56 9999999999999999999  12 3 43 34 54 34 54 34 54 657  5 45 554 12321 12321  65 76 45 45 34 53  546 565 65645 6 65 65 35 65  565 565 | 9999999999999999999 1  12321 4  565 3 |

Cho một số nguyên dương không quá 200 chữ số. Mỗi bước tách số nguyên thành hai nửa: **nửa đầu** là n/2 chữ số đầu tiên, **nửa sau** là phần còn lại (trong đó n là số chữ số của số ban đầu, nếu n lẻ thì phép chia 2 sẽ tính phần nguyên). Sau đó thực hiện tính tổng của hai nửa này.

Lặp lại các bước trên cho đến khi kết quả chỉ còn là số có 1 chữ số.

Hãy thực hiện tính toán và in kết quả của từng bước.

**Input**

Chỉ có một số nguyên dương không quá 200 chữ số.

**Output**

Ghi ra kết quả từng bước, mỗi bước trên một dòng. Dừng lại khi kết quả chỉ còn 1 chữ số.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 123456 | 579  84  12  3 |

Cho dãy số A[] có n phần tử. Hãy sắp xếp các số chẵn trong dãy theo thứ tự tăng dần và các số lẻ theo thứ tự giảm dần.

In ra dãy kết quả đã sắp xếp trong đó vị trí số chẵn và vị trí số lẻ không thay đổi so với dãy ban đầu.

**Input**

Dòng đầu ghi số n (1 < n ≤ 1000)

Các dòng tiếp theo ghi đủ n số của dãy A[], các số đều nguyên dương và không quá 1000.

**Output**

Ghi ra dãy kết quả đã sắp xếp trong đó các vị trí của số chẵn và số lẻ không thay đổi.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 10  1 2 3 4 5 6 7 7 9 6 | 9 2 7 4 7 6 5 3 1 6 |

Giả sử giá trị của một ma trận là hiệu giữa tổng các số trên đường chéo chính và tổng các số trên đường chéo phụ. Cho ma trận A kích thước N \* N, hãy tìm ma trận con của A sao cho ma trận con đó có giá trị lớn nhất.

**Input**

Dòng đầu ghi số N (2 ≤ N ≤ 400)

N dòng tiếp theo ghi ma trận A. Các số trong đoạn [-1000, 1000].

**Output**

Ghi ra giá trị lớn nhất tìm được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  9 -2 -8 0  -6 -2 0 -9  4 -5 6 1  1 3 4 9 | 26 |

*(Giới hạn thời gian chạy: 10 giây)*

Hệ thống quản lý đào tạo của PTIT đang gặp một vấn đề về bảo mật. Do sự cố này, các account bị đổi thành tên viết liền của các sinh viên. Và chỉ cần đánh một chuỗi kí tự có chứa mật khẩu là có thể đăng nhập vào hệ thống. Chẳng hạn sinh viên A có mật khẩu là “abcd”, nếu ai đó đăng nhập với tài khoản là tên của A, mật khẩu “abcdef” hay “aaaabcd” đều được chấp nhận.

Nhân cơ hội này, rất nhiều bạn sinh viên đã cố gắng hack vào tài khoản của những người khác. Cho biết danh sách mật khẩu của tất cả các user, bài toán đặt ra là hãy xác định xem có nhiều nhất bao nhiêu trường hợp user này có thể login vào user khác?

**Input:**

Dòng đầu tiên là số nguyên N (1 ≤  N ≤  100 000).

N dòng tiếp theo, mỗi dòng chứa mật khẩu của một user, có độ dài không quá 10 kí tự và chỉ gồm các kí tự thường.

**Output:**

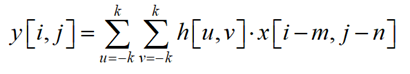
In ra một số nguyên là đáp án đáp án tìm được.

**Ví dụ:**

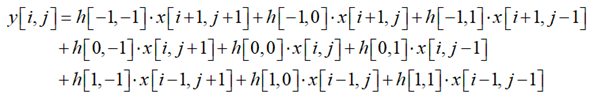
|  |  |
| --- | --- |
| **Test 1** | **Test 2** |
| Input:  3 aaa  aa  abb    Output:  1 | Input:  3  x x xy    Output:  4 |

Giải thích test 2: User 1 có thể login vào user 2 và ngược lại. User 3 có thể login vào tài khoản của user 1 và 2.

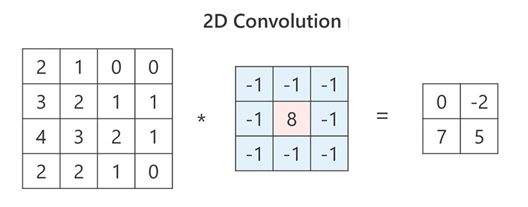
Phép tích chập (convolution) là kỹ thuật quan trọng trong xử lý ảnh. Kết quả phép tích chập giữa ma trận x[] và ma trận kernel h[] được xác định bằng công thức:



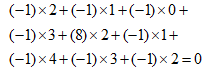
Trong đó ma trận kernel có kích thước bằng 2k+1. Với kernel 3x3 thì -1 ≤u,v≤1, do đó, giá trị các phần tử của ma trận kết quả có dạng:



Cho ma trận ảnh và ma trận kernel 3x3. Nhiệm vụ của bạn là hãy thực hiện phép nhân tích chập của 2 ma trận, sau đó tính **tổng tất cả các phần tử của ma trận thu được.**



Giải thích test: Vị trí ô đầu tiên của ma trận kết quả:



**Input:**

* Dòng đầu tiên là số lượng bộ test T (T≤ 20).
* Mỗi test bắt đầu bởi hai số nguyên N và M. (3≤N,M≤300).
* Kế tiếp là N dòng, mỗi dòng gồm M số nguyên mô tả ma trận ảnh.
* 3 dòng tiếp theo, mỗi dòng gồm 3 số nguyên mô tả ma trận kernel.
* Giá trị các phần tử của hai ma trận có giá trị tuyệt đối không vượt quá 100.

**Output:**

Với mỗi test, hãy in ra **tổng các phần tử của ma trận mới tìm được.**

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4 4  2 1 0 0  3 2 1 1  4 3 2 1  2 2 1 0  -1 -1 -1  -1 8 -1  -1 -1 -1  3 3  1 2 3  4 5 6  7 8 9  1 1 1  1 1 1  1 1 1 | 10  45 |

Cho dãy số A[], B[] và C[] là dãy không giảm và có lần lượt N, M, K phần tử. Nhiệm vụ của bạn là hãy tìm các phần tử chung của 3 dãy số này.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test gồm số nguyên N, M và K (1≤ N, M, K ≤ 100 000).

Dòng tiếp theo gồm N số nguyên A[i], rồi M số nguyên B[i] và K số nguyên C[i].

(0 ≤ A[i], B[i], C[i] ≤ 109).

**Output:**

Với mỗi test, in ra trên một dòng là đáp án thu được. Nếu không tìm được đáp án, in ra “NO”.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 3  6 5 8  1 5 10 20 40 80  5 7 20 80 100  3 4 15 20 30 70 80 120  3 5 4  1 5 5  3 4 5 5 10  5 5 10 20  3 3 3  1 2 3  4 5 6  7 8 9 | 20 80  5 5  NO |

Tại sân bay, mọi người đang làm thủ tục để check in. Có tất cả N vị khách. Vị khách thứ i tới làm thủ tục tại thời điểm T[i] và cần D[i] thời gian để check-in xong.

Các bạn hãy xác định xem thời điểm nào tất cả các vị khách làm xong thủ tục để lên máy bay?

**Input**

Dòng đầu tiên là số nguyên dương N (N ≤ 100).

N dòng tiếp theo, mỗi dòng gồm 2 số nguyên cho biết thời điểm đến của vị khách thứ i và thời gian vị khách này làm xong thủ tục check in. Các giá trị này không vượt quá 106.

**Output**

In ra đáp án tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2 1  8 3  5 7 | 15 |

Giải thích test:

Vị khách đầu tiên tới lúc t = 2 và mất 1 đơn vị thời gian để check in. Vị khách thứ 2 tới lúc t = 5, và làm xong thủ tục tại thời điểm t = 12. Vị khách thứ 3 tới lúc t = 8, nhưng phải chờ tới thời điểm t = 12 để check in, hoàn thành tại thời điểm t = 15.

Cho một số nguyên không âm N được biểu diễn như một xâu ký tự. Hãy sử dụng nhiều nhất một phép đổi chỗ các chữ số trong N sao cho ta nhận được số lớn nhất nhỏ hơn N. Phép biến đổi có số 0 cho số đầu tiên sẽ không được chấp nhận. Ví dụ số N=354 thì số lớn nhất nhỏ hơn N được tạo ra là 345. Số 100 sẽ không có phép biến đổi vì số 010 có số 0 đứng đầu.

**Input:**

Dòng đầu tiên đưa vào số lượng test T (T≤100).

Những dòng kế tiếp đưa vào các test. Mỗi bộ test là một xâu ký tự bao gồm các ký tự số. Độ dài tối đa là 1000.

**Output:**

Với mỗi test in ra số nguyên lớn nhất tìm được trên một dòng. Nếu không tồn tại đáp án, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 4  354  999  100  11101 | 345  -1  -1  11011 |

Cat doi

Tinh tong nho nhat va lon nhat