CA001 - [KIỂM TRA CHẴN LẺ](https://code.ptit.edu.vn/student/question/CA001)

Cho một số nguyên dương N không quá 5 chữ số, hãy kiểm tra và in ra số đó chẵn hay lẻ. Nếu chẵn ghi ra chữ CHAN, nếu ngược lại ghi ra chữ LE.

**Input**

Chỉ có một dòng ghi số N

**Output**

Ghi ra kết quả trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | CHAN |
| 9999 | LE |

Time limit: 1s

Memory limit: 65536 Kb

CA002 - [TÍNH TỔNG PHÂN THỨC](https://code.ptit.edu.vn/student/question/CA002)

Nhập số nguyên dương N (1 < N < 10000).

Viết chương trình tính tổng:

* S = 1 + 1/3 + 1/5 + … + 1/N nếu N lẻ
* S = 1/2 + 1/4 + 1/6 + … + 1/N nếu N chẵn

Kết quả được in ra với 6 chữ số phần thập phân.

**Input**

Dòng đầu ghi số bộ test, không quá 10.

Mỗi test ghi một số N

**Output**

Với mỗi bộ test, ghi ra kết quả trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  10  15 | 1.141667  2.021800 |

Time limit: 1s

Memory limit: 65536 Kb

CA003 - [TÌM SỐ LỚN NHẤT](https://code.ptit.edu.vn/student/question/CA003)

Cho luồng dữ liệu văn bản chỉ bao gồm các số nguyên dương không quá 18 chữ số. Hãy tìm và in ra giá trị số lớn nhất.

Chú ý: bạn cần đọc đến hết luồng dữ liệu vào.

**Input**

Luồng văn bản với không quá 1 triệu số nguyên dương. Các số đều không quá 18 chữ số.

**Output**

Ghi ra số lớn nhất tìm được

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 23 545 64334 465 3  54 34 999 11111111111 | 11111111111 |

Time limit: 1s

Memory limit: 65536 Kb

CA004 - [LIỆT KÊ SỐ KHÁC NHAU](https://code.ptit.edu.vn/student/question/CA004)

Cho dãy số A[] chỉ bao gồm các số nguyên dương không quá 1000. Hãy liệt kê các số khác nhau trong dãy theo thứ tự tăng dần.

**Input**

Dòng đầu ghi số n là số phần từ của dãy (không quá 100).

Dòng thứ 2 ghi n số của mảng.

**Output**

Ghi ra các số khác nhau trong dãy A[] theo thứ tự tăng dần, mỗi số cách nhau một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  3 7 9 3 6 13 7 7 9 3 | 3 6 7 9 13 |

Time limit: 1s

Memory limit: 65536 Kb

CA005 - [K CHỮ SỐ CUỐI](https://code.ptit.edu.vn/student/question/CA005)

Cho số nguyên dương N có ít nhất 3 chữ số nhưng không quá 9 chữ số.

Với một giá trị K nhỏ hơn số chữ số của N, hãy kiểm tra xem số tạo bởi K chữ số cuối cùng của N có phải là số thuận nghịch hay không.

**Input**

Chỉ có 1 dòng ghi số nguyên dương N (có từ 3 đến 9 chữ số), một khoảng trống, sau đó đến số nguyên dương K (đảm bảo nhỏ hơn số chữ số của N).

**Output**

Nếu số tạo bởi K chữ số cuối cùng của N là số thuận nghịch thì ghi ra YES, nếu không phải ghi ra NO

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 16690878 3 | YES |
| 14435639 3 | NO |

Time limit: 1s

Memory limit: 65536 Kb

CA006 - [K CHỮ SỐ ĐẦU TIÊN](https://code.ptit.edu.vn/student/question/CA006)

Cho số nguyên dương N có ít nhất 3 chữ số nhưng không quá 18 chữ số.

Với một giá trị K nhỏ hơn số chữ số của N, hãy tìm số X được tạo bởi K chữ số đầu tiên của N. Sau đó tính giá trị phần nguyên của X/K.

**Input**

Chỉ có 1 dòng ghi số nguyên dương N (có từ 3 đến 18 chữ số), một khoảng trống, sau đó đến số nguyên dương K (đảm bảo nhỏ hơn số chữ số của N).

**Output**

Ghi ra kết quả tính được trên 1 dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 123456789012 4 | 308 |

*Giải thích ví dụ: 4 số đầu của N là 1234, kết quả bằng 1234/4 = 308 (chỉ lấy phần nguyên)*

Time limit: 1s

Memory limit: 65536 Kb

CA007 - [TÁCH TRÁI PHẢI](https://code.ptit.edu.vn/student/question/CA007)

Cho số nguyên dương N không quá 18 chữ số. Nếu số chữ số của N là chẵn thì ta có thể tách thành hai nửa trái và phải có số chữ số bằng nhau.

Hãy tính ước chung lớn nhất của hai nửa trái và phải của số N.

**Input**

Dòng đầu ghi số bộ test (không quá 10)

Mỗi bộ test ghi một số N, không quá 18 chữ số.

**Output**

Nếu số chữ số là lẻ thì ghi ra INVALID

Nếu số chữ số là chẵn thì in ra ước chung lớn nhất của hai nửa trái và phải.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  7  1220  1234567 | INVALID  4  INVALID |

Time limit: 1s

Memory limit: 65536 Kb

CA008 - [SỐ MAY MẮN](https://code.ptit.edu.vn/student/question/CA008)

Chữ số 4 và chữ số 7được xem là các chữ số may mắn.

Cho số nguyên dương N có không quá 18 chữ số. Hãy đếm xem số chữ số 4 cộng với số chữ số 7 trong N có phải bằng 4 hay bằng 7 hay không.

**Input**

Chỉ có số N

**Output**

Ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 40047 | NO |
| 7747774 | YES |
| 1000000000000000000 | NO |

Time limit: 1s

Memory limit: 65536 Kb

CA009 - [SỐ TAM PHÂN](https://code.ptit.edu.vn/student/question/CA009)

Một số được gọi là “tam phân” nếu chỉ có các chữ số 0,1,2. Nhập vào một số nguyên dương không quá 9 chữ số, hãy kiểm tra xem đó có phải số tam phân hay không. Dòng đầu là số bộ test, mỗi dòng tiếp theo ghi một số cần kiểm tra. Nếu đúng in ra YES, nếu sai in ra NO.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1214  10210221  22222222 | NO  YES  YES |

Time limit: 1s

Memory limit: 65536 Kb

CA010 - [CHỮ SỐ NGUYÊN TỐ](https://code.ptit.edu.vn/student/question/CA010)

Một số nguyên dương được gọi là ưu thế nguyên tố nếu thỏa mãn cả hai điều kiện:

* Số chữ số của nó là một số nguyên tố
* Số lượng chữ số nguyên tố nhiều hơn số lượng chữ số không nguyên tố

Viết chương trình kiểm tra một số nguyên có thỏa mãn ưu thế nguyên tố hay không.

**Input**

* Dòng đầu ghi số bộ test, không quá 20.
* Mỗi bộ test ghi số nguyên dương N, ít nhất 3 chữ số nhưng không quá 500 chữ số

**Output**

Với mỗi bộ test, ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1234567  22334455667  23400300489898989 | YES  YES  NO |

Time limit: 1s

Memory limit: 65536 Kb

CA011 - [SỐ CHIA HẾT CHO 3](https://code.ptit.edu.vn/student/question/CA011)

Cho số nguyên dương N có thể rất lớn nhưng không quá 500 chữ số.

Hãy kiểm tra xem N có chia hết cho 3 hay không.

**Input**

Dòng đầu ghi số bộ test (không quá 20).

Mỗi test ghi số N (không quá 500 chữ số)

**Output**

Với mỗi bộ test, ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  12341  123456789123456789 | NO  YES |

 Time limit: 1s

Memory limit: 65536 Kb

CA012 - [TẬP HỢP SỐ BẰNG NHAU](https://code.ptit.edu.vn/student/question/CA012)

Cho dãy số a[] có n phần tử và dãy số b[] có m phần tử là các số nguyên dương nhỏ hơn 1000. Gọi tập hợp A là tập các số khác nhau trong a[], tập hợp B là tập các số khác nhau trong b[].

Tập A và tập B được coi là bằng nhau nếu số phần tử bằng nhau và tất cả các giá trị số từ nhỏ đến lớn đều bằng nhau từng đôi một. Khi A = B ta cũng có thể kết luận là hai dãy a[] và b[] chứa các số giống nhau.

Hãy kiểm tra xem A có bằng B hay không?

**Input**

Dòng đầu ghi 2 số n và m (1 < n,m <100).

Dòng thứ 2 ghi n số của a[].

Dòng thứ 3 ghi m số của b[].

Các số đều dương và nhỏ hơn 1000.

**Output**

Ghi ra kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12 18  1 2 3 4 5 1 2 3 5 4 1 2  1 1 1 1 1 1 1 1 1 2 3 4 5 5 5 5 5 5 | YES |

Time limit: 1s

Memory limit: 65536 Kb

CA013 - [LIỆT KÊ SỐ NGUYÊN TỐ TRONG DÃY](https://code.ptit.edu.vn/student/question/CA013)

Cho dãy số nguyên dương A[] có N phần tử. Hãy viết chương trình liệt kê các số nguyên tố khác nhau và số lần xuất hiện của số đó trong dãy ban đầu.

Các số được liệt kê theo thứ tự xuất hiện.

**Input**

Dòng đầu ghi số N (không quá 500).

Dòng sau ghi N số của dãy (không quá 6 chữ số).

**Output**

Ghi ra các số nguyên tố khác nhau trong dãy theo thứ tự xuất hiện và số lần xuất hiện. Mỗi số liệt kê trên 1 dòng.

**Ví dụ**

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

Time limit: 1s

Memory limit: 65536 Kb

CA014 - [CÁC CHỮ SỐ XUẤT HIỆN TRONG DÃY](https://code.ptit.edu.vn/student/question/CA014)

Cho mảng A[] gồm n phần tử. Nhiệm vụ của bạn là đưa ra mảng đã được sắp xếp bao gồm các chữ số của mỗi phần tử trong A[]. Ví dụ A[] = {110, 111, 112, 113, 114 }ta có kết quả là {0, 1, 2, 3, 4}.

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A[i] ; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n ≤107; 0≤ A[i] ≤1016.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| **Input:** | **Output:** |
| 2 3 131 11 48 4 111 222 333 446 | 1 3 4 8 1 2 3 4 6 |

Time limit: 1s

Memory limit: 65536 Kb

CA015 - [LIỆT KÊ CÁC SỐ CÓ HAI CHỮ SỐ TĂNG DẦN](https://code.ptit.edu.vn/student/question/CA015)

Cho một dãy ký tự số không quá 1000 chữ số và không có chữ số 0.

Người ta lần lượt lấy ra mỗi lần 2 chữ số tính từ trái sang phải. Nếu bước cuối cùng không đủ hai chữ số thì bỏ qua chữ số đó. Kết quả sẽ được một dãy số nguyên dương A[] chỉ bao gồm các số có hai chữ số.

Hãy liệt kê **các số khác nhau xuất hiện trong A[]** theo **thứ tự tăng dần**.

**Input**

Chỉ có một dòng ghi dãy ký tự số (độ dài không quá 1000). Dữ liệu vào đảm bảo không có chữ số 0.

**Output**

Ghi ra lần lượt các số khác nhau xuất hiện trong dãy A[] theo thứ tự tăng dần, mỗi số viết cách nhau một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 124356141111434356149 | 11 12 14 43 56 |

Time limit: 1s

Memory limit: 65536 Kb

CA016 - [KIỂM TRA NGUYÊN TỐ CÙNG NHAU](https://code.ptit.edu.vn/student/question/CA016)

Cho hai số nguyên dương N và M, cả hai đều không quá 9 chữ số. Hãy kiểm tra xem hai số này có nguyên tố cùng nhau hay không.

**Input**

Dòng đầu ghi số T là số bộ test (T < 10).

T dòng tiếp theo, mỗi dòng ghi 2 số nguyên dương N và M, không quá 9 chữ số

**Output**

Với mỗi bộ test, ghi ra YES hoặc NO tuỳ thuộc kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  123 456  123 49 | NO  YES |

Time limit: 1s

Memory limit: 65536 Kb

CA017 - [ĐẾM CÁC SỐ NGUYÊN TỐ CÙNG NHAU](https://code.ptit.edu.vn/student/question/CA017)

Một cặp số nguyên (a,b) được gọi là nguyên tố cùng nhau nếu ước số chung lớn nhất của a và b bằng 1.

Cho số nguyên dương N, hãy đếm xem có bao nhiêu số nguyên dương nhỏ hơn hoặc bằng N thoả mãn nguyên tố cùng nhau với N.

**Input**

Dòng đầu ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương N (1 ≤ N ≤ 109).

**Output**

Ghi ra số lượng các số nhỏ hơn hoặc bằng N thoả mãn nguyên tố cùng nhau với N.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 5  100  200  155  210  985 | 40  80  120  48  784 |

Time limit: 5s

Memory limit: 65536 Kb

CA018 - [LIỆT KÊ NGUYÊN TỐ CÙNG NHAU](https://code.ptit.edu.vn/student/question/CA018)

Một cặp số nguyên dương (a,b) được gọi là nguyên tố cùng nhau nếu a và b có ước chung lớn nhất bằng 1.

Cho hai số nguyên dương N và K trong đó 10 < N < 10000; 1 < K < 6.

Hãy liệt kê các số có K chữ số thỏa mãn nguyên tố cùng nhau với N.

**Input**

Chỉ có một dòng ghi hai số N và K

**Output**

Ghi ra lần lượt các số thỏa mãn theo thứ tự từ nhỏ đến lớn. Mỗi dòng ghi 10 số.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 123 2 | 10 11 13 14 16 17 19 20 22 23  25 26 28 29 31 32 34 35 37 38  40 43 44 46 47 49 50 52 53 55  56 58 59 61 62 64 65 67 68 70  71 73 74 76 77 79 80 83 85 86  88 89 91 92 94 95 97 98 |

Time limit: 1s

Memory limit: 65536 Kb

CA019 - [SỐ NHỎ NHẤT CÒN THIẾU](https://code.ptit.edu.vn/student/question/CA019)

Cho dãy số A[] có N phần tử là các số nguyên dương khác nhau. Hãy tìm số nhỏ nhất còn thiếu trong dãy số đó.

**Input**

Dòng đầu ghi số N (1 <= N <= 30000).

Dòng tiếp theo ghi N số của dãy A (1 <= A[i] <= 30000).

**Output**

Ghi ra số nhỏ nhất còn thiếu nếu có.

(khi dãy số đầy đủ các số từ 1 đến N thì số nhỏ nhất còn thiếu sẽ là N+1).

**Ví dụ**

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

Time limit: 1s

Memory limit: 65536 Kb

CA020 - [SỐ LẦN XUẤT HIỆN LỚN NHẤT](https://code.ptit.edu.vn/student/question/CA020)

Cho dãy số A[] có N phần tử là các số nguyên dương nhỏ hơn 1000.

Hãy đếm số lần xuất hiện của các số khác nhau trong dãy. Sau đó tìm số xuất hiện nhiều lần nhất và in ra số đó. Chú ý: trong trường hợp có nhiều số có cùng số lần xuất hiện lớn nhất thì cần in ra tất cả các số đó theo thứ tự tăng dần.

**Input**

Dòng đầu ghi số N ( 1 < N < 1000).

Dòng 2 ghi N số nguyên dương trong dãy A (0 < A[i] < 1000)

**Output**

Dòng đầu ghi số lần xuất hiện lớn nhất tìm được.

Dòng thứ 2 ghi lần lượt các số xuất hiện nhiều lần nhất theo thứ tự tử nhỏ đến lớn.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9  3 4 3 1 6 4 4 1 1 | 3  1 4 |

Time limit: 1s

Memory limit: 65536 Kb

CA021 - [SỐ NGUYÊN TỐ ĐẸP](https://code.ptit.edu.vn/student/question/CA021)

Cho sô nguyên dương N, không quá 6 chữ số.

N được gọi là số nguyên tố đẹp nếu thoả mãn cả hai điều kiện:

* N là số nguyên tố
* Tổng chữ số của N chia hết cho 10

Hãy kiếm tra xem N có thoả mãn là số nguyên tố đẹp hay không?

**Input**

Dòng đầu ghi số T là số bộ test (T < 10).

T dòng tiếp theo, mỗi dòng ghi số nguyên dương N, không quá 6 chữ số

**Output**

Với mỗi bộ test, ghi ra YES hoặc NO tuỳ thuộc kết quả kiểm tra

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  37  229 | YES  NO |

 Time limit: 1s

Memory limit: 65536 Kb

CA022 - [SỐ TĂNG GIẢM](https://code.ptit.edu.vn/student/question/CA022)

Một số nguyên dương được gọi là số tăng giảm nếu thỏa mãn các điều kiện:

* Có từ 3 chữ số trở lên
* Tìm ra một vị trí trong dãy chữ số sao cho từ bên trái đến vị trí đó thỏa mãn thứ tự tăng dần (tăng chặt) còn từ vị trí đó đến hết thì thỏa mãn thứ tự giảm dần (giảm chặt).

Viết chương trình kiểm tra một số có phải số tăng giảm hay không.

**Input**

Dòng đầu ghi số bộ test. Mỗi bộ test viết trên một dòng số nguyên dương N không quá 18 chữ số

**Output**

Với mỗi bộ test, ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 3  12342  23342  5678961 | YES  NO  YES |

 Time limit: 1s

Memory limit: 65536 Kb

CA023 - [GIÁ TRỊ NHỊ PHÂN](https://code.ptit.edu.vn/student/question/CA023)

Cho các giá trị nhị phân 0 và 1.

Hãy đếm xem có bao nhiêu cặp giá trị khác nhau đứng cạnh nhau.

**Input**

Dòng 1 ghi số N là số giá trị nhị phân (không quá 100).

Dòng 2 ghi N giá trị nhị phân.

**Output**

Ghi ra kết quả bài toán.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  1 0 0 1 1 1 | 2 |

Time limit: 1s

Memory limit: 65536 Kb

CA024 - [SẮP XẾP THEO TỔNG CHỮ SỐ](https://code.ptit.edu.vn/student/question/CA024)

Cho dãy số A[] có N phần tử đều là các số nguyên dương, không quá 9 chữ số.

Hãy sắp xếp dãy số theo tổng chữ số tăng dần. Nếu tổng chữ số bằng nhau thì số nào nhỏ hơn sẽ viết trước.

**Input**

Dòng đầu ghi số bộ test (không quá 10)

Mỗi bộ test gồm 2 dòng:

* Dòng đầu là số N (N < 100)
* Dòng thứ 2 ghi N số của mảng A[], các số đều nguyên dương và không quá 9 chữ số.

**Output**

Với mỗi bộ test, ghi trên một dòng dãy số kết quả.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  8  143 43 22 99 7 9 1111 10000000 | 10000000 22 1111 7 43 143 9 99 |

Time limit: 1s

Memory limit: 65536 Kb

CA025 - [TỔNG CHỮ SỐ CHIA HẾT CHO 4](https://code.ptit.edu.vn/student/question/CA025)

Cho số nguyên dương N.

Hãy tìm số nguyên dương M là số nhỏ nhất lớn hơn hoặc bằng N và tổng chữ số của M chia hết cho 4.

**Input**

Chỉ có duy nhất 1 số nguyên dương N, không quá 1000.

**Output**

Ghi ra giá trị tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 99 | 103 |

 Time limit: 1s

Memory limit: 65536 Kb

CA026 - [BỘI SỐ CHUNG NHỎ NHẤT](https://code.ptit.edu.vn/student/question/CA026)

Cho số nguyên dương N có đúng 4 chữ số.

Ta đã biết rằng có thể lấy 2 số cuối của N bằng phép chia dư cho 100.

Hãy thực hiện tách N thành 2 phần:

* Số a tạo bởi 2 chữ số đầu của N
* Số b tạo bởi 2 chữ số cuối của N

Tính bội số chung nhỏ nhất của a và b.

**Input**

Chỉ có 1 dòng ghi số N có đúng 4 chữ số.

**Output**

Ghi ra kết quả tính được trên 1 dong

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1008 | 40 |

Time limit: 1s

Memory limit: 65536 Kb

CA027 - [SỐ CHÍNH PHƯƠNG](https://code.ptit.edu.vn/student/question/CA027)

Cho số nguyên dương N có ít nhất 2 chữ số nhưng không quá 9 chữ số.

Hãy kiểm tra xem có thể ghép dần các chữ số của N, tính từ chữ số cuối cùng, để tạo nên các số chính phương hay không?

**Input**

Chỉ có 1 dòng ghi số N có từ 2 đến 9 chữ số.

Dữ liệu đảm bảo N không có chữ số 0

**Output**

Nếu có thể tạo nên ít nhất 1 số chính phương thì ghi lần lượt các số chính phương tạo được theo thứ tự từ nhỏ đến lớn, mỗi số 1 dòng.

Nếu không thể tạo được số chính phương nào thì ghi ra NO

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 249361 | 1  361 |
| 3763 | NO |

*Giải thích ví dụ 1: các số tạo được là:*

*1****: là số chính phương***

*61                 : không phải số chính phương*

*361****: là số chính phương***

*9361             : không phải số chính phương*

*49361           : không phải số chính phương*

*249361         : không phải số chính phương*

Time limit: 1s

Memory limit: 65536 Kb

CA028 - [TỔNG CHẴN](https://code.ptit.edu.vn/student/question/CA028)

Cho dãy số nguyên có N phần tử, chỉ số tính từ 1.

Hãy tính tổng các số nằm ở vị trí chẵn của dãy.

**Input**

Dòng đầu ghi số N (không quá 100.000).

Dòng tiếp theo ghi N số của dãy, các giá trị có trị tuyệt đối không quá 100.000.

**Output**

Ghi ra kết quả tính được trên 1 dòng.

**Ví dụ**

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

Time limit: 1s

Memory limit: 65536 Kb

CA029 - [TỔNG FIBONACCI](https://code.ptit.edu.vn/student/question/CA029)

Dãy số Fibonacci được định nghĩa như sau:

F[1] = 1

F[2] = 1

F[i] = F[i-1] + F[i-2] với i > 2

Hãy tính tổng của các số từ vị trí a đến vị trí b trong dãy số F[].

**Input**

Chỉ có một dòng ghi hai số a và b (1 < a < b < 60).

**Output**

Ghi ra kết quả tính được trên 1 dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 5 | 10 |

*Giải thích ví dụ: 5 số đầu tiên của dãy F[] là 1, 1, 2, 3, 5*

Time limit: 1s

Memory limit: 65536 Kb

CA030 - [SẮP XẾP NGUYÊN TỐ](https://code.ptit.edu.vn/student/question/CA030)

Cho dãy số nguyên dương A[] có N phần tử. Các giá trị trong dãy không quá 1000.

Hãy sắp xếp các số nguyên tố trong dãy theo thứ tự tăng dần. Các giá trị không nguyên tố vẫn giữ nguyên vị trí như lúc đầu.

Xem ví dụ để hiểu rõ hơn yêu cầu bài toán.

**Input**

Dòng đầu ghi số N (1 < N < 100), dòng thứ 2 ghi N số của dãy A[].

**Output**

Ghi ra dãy số kết quả trên một dòng.

**Ví dụ**

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

Time limit: 1s

Memory limit: 65536 Kb

CA031 - [KIỂM TRA MA TRẬN ĐỐI XỨNG](https://code.ptit.edu.vn/student/question/CA031)

Ma trận vuông là ma trận mà các phần tử có vị trí đối xứng với nhau qua đường chéo chính có giá trị bằng nhau.

Viết chương trình kiểm tra một ma trận vuông có phải là ma trận đối xứng qua đường chéo chính hay không.

Yêu cầu:

**Input:**

Dòng đầu tiên ghi số bộ test

Mỗi bộ test bao gồm:

Dòng đầu tiên ghi kích thước của ma trận vuông n

n dòng tiếp theo, mỗi dòng có n số cách nhau bởi dấu cách.

**Ouput:**

YES nếu là ma trận đối xứng, NO nếu ma trận không đối xứng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Ouput** |
| 2  2  1 2  2 3  3  1 2 3  2 1 3  3 2 1 | YES  NO |

Time limit: 2s

Memory limit: 65536 Kb

CA032 - [TÍCH MA TRẬN](https://code.ptit.edu.vn/student/question/CA032)

Viết chương trình tính tích hai ma trận A cỡ n\*m và ma trận B cỡ m\*p.

Với 1 < n,m,p < 50. Các giá trị trong ma trận đều nguyên dương và không vượt quá 1000.

**Input**

Dòng đầu ghi 3 số n,m,p

n dòng tiếp theo ghi ma trận A

m dòng tiếp theo ghi ma trận B

**Output**

Ghi ra ma trận tích

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 4 3  1 2 3 4  4 2 3 1  2 4 1 3  1 1 1  2 2 2  3 3 3  4 4 4 | 30 30 30  21 21 21  25 25 25 |

Time limit: 2s

Memory limit: 65536 Kb

CA033 - [MA TRẬN NHỊ PHÂN](https://code.ptit.edu.vn/student/question/CA033)

Cho ma trận A[] có N hàng và 3 cột, trong đó các vị trí là các giá trị nhị phân (0 hoặc 1). Hãy đếm xem có bao nhiêu hàng mà số lượng số 1 nhiều hơn số lượng số 0.

**Input**

Dòng đầu ghi số nguyên dương N (không quá 1000).

N dòng tiếp theo, mỗi dòng ghi 3 giá trị nhị phân.

**Output**

Ghi ra số dòng mà số lượng số 1 nhiều hơn số lượng số 0.

**Ví dụ**

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

Time limit: 2s

Memory limit: 65536 Kb

CA034 - [KIỂM TRA NGUYÊN TỐ](https://code.ptit.edu.vn/student/question/CA034)

Cho ma trận A[] cỡ N\*M chỉ bao gồm các số nguyên dương không quá 1000. Hãy kiểm tra các số trong ma trận, nếu giá trị nào là số nguyên tố thì thay thế bằng số 1, không phải thì thay thế bằng số 0.

**Input**

Dòng đầu ghi 2 số N và M là kích thước ma trận (1 < N,M < 20)

N dòng tiếp theo mỗi dòng có M số mô tả ma trận

**Output**

Ghi ra ma trận kết quả

**Ví dụ**

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

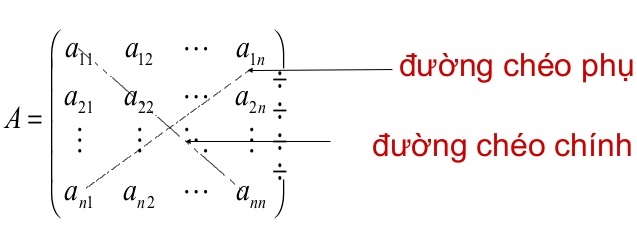
Time limit: 2s

Memory limit: 65536 Kb

CA035 - [ĐƯỜNG CHÉO CHÍNH CỦA MA TRẬN](https://code.ptit.edu.vn/student/question/CA035)

Cho ma trận vuông cấp N\*N chỉ bao gồm các số nguyên dương.

Với đường chéo chính, ta sẽ chia ma trận thành 2 nửa, được gọi là nửa trên và nửa dưới của đường chéo chính (không tính các phần tử nằm trên đường chéo chính).



Độ chênh lệch của ma trận được tính bằng trị tuyệt đối khi lấy **tổng giá trị các phần tử ở nửa trên** trừ đi **tổng giá trị các phần tử ở nửa dưới**.

Nhập thêm một giá trị K gọi là *ngưỡng cân đối của ma trận*.  Trong trường hợp độ chênh lệch không quá K thì ma trận được coi là cân đối, nếu lớn hơn K thì không cân đối.

Hãy xác định độ chênh lệch và tính cân đối của ma trận.

**Input**

Dòng đầu ghi số N (2 < N < 50)

N dòng tiếp theo ghi các giá trị của ma trận, các số đều nguyên dương và không quá 1000.

Dòng cuối ghi số K (0 < K <100)

**Output**

Dòng đầu ghi chữ YES hoặc NO

Dòng thứ 2 ghi ra giá trị độ chênh lệch của ma trận

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  2 8 10 6 7  6 3 2 6 9  10 2 6 2 8  9 9 7 9 8  9 6 5 6 9  5 | YES  3 |

Time limit: 2s

Memory limit: 65536 Kb

CA036 - [SỐ NGUYÊN TỐ LỚN NHẤT](https://code.ptit.edu.vn/student/question/CA036)

Cho ma trận A cỡ N\*M chỉ bao gồm các số nguyên dương.

Hãy tìm số nguyên tố lớn nhất trong ma trận và các vị trí có giá trị bằng số nguyên tố lớn nhất đó.

**Input**

Dòng đầu ghi hai số N và M (1 < N, M < 50)

Tiếp theo là N dòng ghi các giá trị của ma trận, không có số nào lớn hơn 1000.

**Output**

Ghi ra giá trị của số nguyên tố lớn nhất. Sau đó lần lượt là các vị trí của số nguyên tố lớn nhất, mỗi vị trí trên một dòng (chỉ số hàng và cột tính từ 0). Các vị trí được liệt kê theo thứ tự từ trái qua phải, từ trên xuống dưới.

Nếu không tìm thấy số nguyên tố nào thì ghi ra NOT FOUND

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **.Output** |
| 6 4  23 21 26 10  13 13 22 14  28 29 28 23  29 19 11 19  16 26 24 21  13 25 21 29 | 29  Vi tri [2][1]  Vi tri [3][0]  Vi tri [5][3] |

Time limit: 2s

Memory limit: 65536 Kb

CA037 - [MA TRẬN XOẮN ỐC TĂNG DẦN](https://code.ptit.edu.vn/student/question/CA037)

Cho ma trận vuông A cỡ N\*N chỉ bao gồm các số nguyên dương không quá 1000. Hãy sắp đặt các giá trị trong ma trận A sao cho các số được điền lần lượt theo kiểu xoắn ốc tăng dần, theo chiều kim đồng hồ.

**Input**

Dòng đầu ghi số N (2 < N < 20).

N dòng tiếp theo ghi ma trận A, các giá trị nguyên dương và không quá 1000.

**Output**

Ghi ra ma trận kết quả

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  3 6 1  8 7 9  4 12 5 | 1 3 4  9 12 5  8 7 6 |

Time limit: 2s

Memory limit: 65536 Kb

CA038 - [BIẾN ĐỔI VỀ MA TRẬN VUÔNG](https://code.ptit.edu.vn/student/question/CA038)

Cho ma trận A kích thước N\*M chỉ bao gồm các số nguyên dương.

Trong trường hợp N # M, hãy biến đổi ma trận A về dạng ma trận vuông theo quy tắc sau:

* Nếu N > M, hãy loại bỏ các **hàng có thứ tự lẻ** trong ma trận ban đầu (thứ tự hàng tính từ 1) cho đến khi N = M. Ví dụ N = 6, M = 4 thì cần loại bỏ hàng thứ 1 và hàng thứ 3.
* Nếu M > N, hãy loại bỏ các **cột có thứ tự chẵn** trong ma trận ban đầu (thứ tự cột tính từ 1). Ví dụ: N = 4, M = 6 thì cần loại bỏ cột thứ 2 và cột thứ 4.

In ra ma trận kết quả sau khi biến đổi.

**Input**

Dòng đầu ghi hai số N và M (1 < N,M < 50).

N dòng tiếp theo ghi các phần tử của ma trận A, các giá trị đều nguyên dương và không quá 1000.

**Output**

Ghi ra ma trận vuông sau khi biến đổi.

**Ví dụ**

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

Time limit: 2s

Memory limit: 65536 Kb