

Biến đổi chuỗi

Cho chuỗi A và chuỗi B độ dài không quá 500. Các ký tự được đánh số từ 0. Cần biến đổi chuỗi A thành chuỗi B bằng ít nhất các phép biến đổi. Có các phép biến đổi như sau:

- **INSERT position character:** chèn ký tự **character** vào vị trí **position**. Ví dụ áp dụng **INSERT 1 p** với chuỗi **abcd** sẽ thu được **apbcd**
- **CHANGE position character:** đổi ký tự tại vị trí **position** thành ký tự **character**. Ví dụ áp dụng **CHANGE 0 x** với chuỗi **abcd** sẽ thu được **xbcd**.
- **DELETE position:** xóa ký tự tại vị trí **position**. Ví dụ áp dụng **DELETE 0** với chuỗi **abcd** thu được **bcd**.
- **SWAP position1 position2:** đổi chỗ hai ký tự tại vị trí **position1**, **position2**. Ví dụ áp dụng **SWAP 0 2** với chuỗi **abcd** sẽ thu được **cbad**.

Input: BDX.IN

- Dòng 1 là chuỗi A gồm các ký tự in hoa và in thường.
- Dòng 2 là chuỗi B gồm các ký tự in hoa và in thường

Output: BDX.OUT

- In ra số phép biến đổi P.
- P dòng tiếp theo ghi ra các phép biến đổi.

Ví dụ:

Input	Output
ABCDEF BDCF	3 DELETE 0 SWAP 1 2 DELETE 3

Bưu điện 1&2

Có N ngôi làng trên một con đường thẳng. Ngôi làng thứ u nằm ở vị trí D_u . Chúng ta cần đưa ra 2 phương án chọn ra K ngôi làng để xây dựng bưu điện ở đó sao cho:

- Tổng khoảng cách của mỗi ngôi làng đến bưu điện gần nó nhất là nhỏ nhất.
- Khoảng cách xa nhất của mỗi ngôi làng đến bưu điện gần nó nhất là nhỏ nhất.

Input POST1.IN và POST2.IN:

- Dòng đầu tiên ghi số N và K . ($1 \leq K \leq N \leq 500$)
- Dòng thứ hai ghi N số D_u tăng dần. Các số trong input không quá 10^6 .

Output 1: POST1.OUT

- Dòng đầu tiên ghi tổng khoảng cách.
- Dòng thứ hai ghi K số là số hiệu của các ngôi làng ta sẽ xây dựng bưu điện.

Output2: POST1.OUT

- Dòng đầu tiên ghi khoảng cách xa nhất.
- Dòng thứ hai ghi K số là số hiệu của các ngôi làng ta sẽ xây dựng bưu điện.

Ví dụ

POST1.IN	POST1.OUT
10 5	9
1 2 3 6 7 9 11 22 44 50	2 5 8 9 10

POST2.IN	POST2.OUT
10 5	3
1 2 3 6 7 9 11 22 44 50	2 6 8 9 10

MAX CLIQUE

Cho đồ thị vô hướng N đỉnh. Một tập con của đồ thị được gọi là clique nếu giữa mọi cặp đỉnh trong tập con đều có cạnh nối. Một tập con S được gọi là max clique nếu không có cách thêm một đỉnh nào chưa thuộc S mà tạo được thành clique mới.

Hãy đếm số lượng max clique độ lớn $1, 2, \dots, N$.

Input: MAXCLIQUE.IN

- Dòng đầu tiên ghi số N . ($N \leq 25$)
- Tiếp theo là N dòng, mỗi dòng ghi N số $0/1$ tương ứng với không có/có cạnh nối giữa hai đỉnh.

Output: MAXCLIQUE.OUT

- Ghi ra N số, số thứ K là số lượng max clique độ lớn K .

MAXCLIQUE.IN	MAXCLIQUE.OUT
6 1 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 0 0 1	1 1 1 0 0 0

Bưu điện 3 & 4

Giống bưu điện 1 & 2 nhưng các ngôi làng không nằm trên đường thẳng mà được nối với nhau bởi các đường đi hai chiều.

Input: POST3.IN và POST4.IN

Dòng một ghi số N ($N \leq 30$).

Tiếp theo là N dòng, mỗi dòng ghi N số thể hiện khoảng cách đường nối trực tiếp giữa hai ngôi làng. Hai ngôi làng không có đường nối ghi số 0.

Output: POST3.OUT và POST4.OUT

Giống bưu điện 1 & 2.

DIVISOR 1&2

Cho 1 số tự nhiên N , gọi A là dãy chứa các ước số của N ($N \bmod A[i] = 0$).

- Bạn hãy viết chương trình sắp xếp lại dãy A trên sao cho dãy đó thỏa mãn tính chất sau: với 2 phần tử liên tiếp $A[i]$ và $A[i+1]$ thì $A[i] = A[i+1] * p$, hoặc $A[i+1] = A[i] * p$ ($1 \leq i < N$), với p là một số nguyên tố.
- Đếm số cách sắp xếp thỏa mãn

Input DIVISOR1.IN DIVISOR2.OUT chứa 1 số nguyên N duy nhất.

Output DIVISOR1.OUT gồm các dòng:

- Dòng thứ 1 ghi số tự nhiên M – số lượng phần tử của dãy A (số ước của N).
- Dòng thứ 2 gồm M số là các phần tử của A sau khi sắp xếp.

Output DIVISOR2.OUT in ra một số duy nhất là số cách sắp xếp.

Giới hạn:

- Với bài DIVISOR1 $2 \leq N \leq 2,000,000,000$, nếu có nhiều lời giải chỉ cần xuất 1 lời giải bất kỳ.
- Với bài DIVISOR2 $2 \leq N \leq 2,000,000$

Ví dụ:

DIVISOR1.INP	DIVISOR1.OUT
20	6 1 2 4 20 10 5