

## ĐỀ LUYỆN TẬP KỲ THI ICPC TUẦN 2

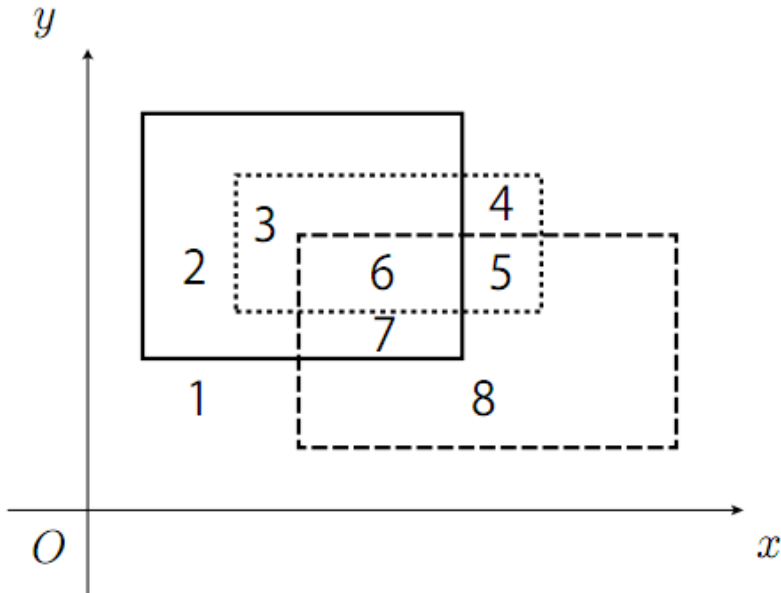
Ngày thi: Chủ nhật, ngày 28 tháng 07 năm 2024. Đề thi gồm có 10 bài

### BÀI A. CÁC HÌNH CHỮ NHẬT

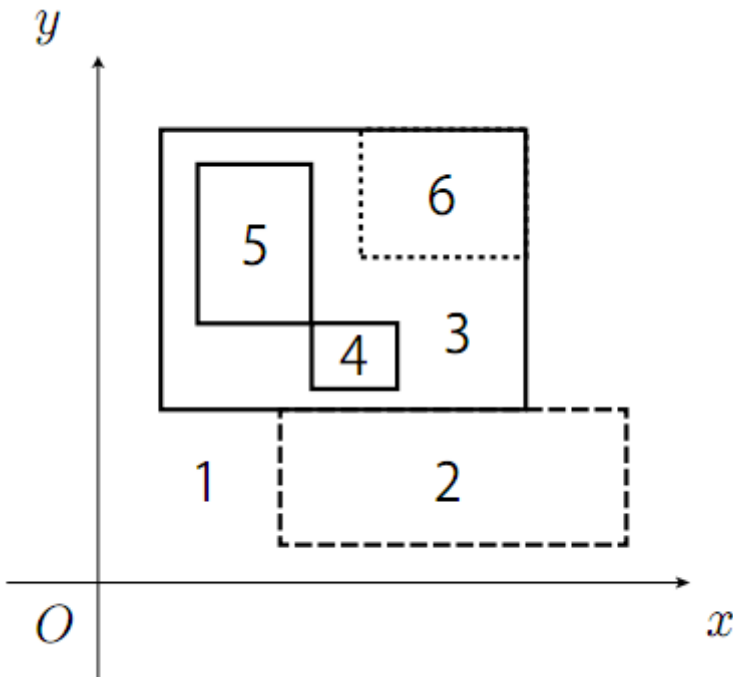
Cho trước  $N$  hình chữ nhật trong hệ trục tọa độ Oxy. Các hình chữ nhật này chia mặt phẳng thành các vùng riêng biệt, mà giữa 2 vùng bất kì không tồn tại đường đi giữa chúng.

Nhiệm vụ của bạn là đếm số vùng riêng biệt đã được tạo ra.

Hình vẽ test 1:



Hình vẽ test 2:



#### Input:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 200$ ).

Mỗi bộ test gồm số nguyên dương  $N$  ( $N \leq 50$ ) là số hình chữ nhật.

$N$  dòng tiếp theo, mỗi dòng gồm 4 số nguyên  $x_A, y_A, x_B, y_B$  là tọa độ của góc trái trên và góc phải dưới của hình chữ nhật thứ  $i$ . (Các tọa độ là các số nguyên không âm và không vượt quá  $10^6$ ).

**Output:**

Với mỗi test, in ra số lượng các vùng riêng biệt được tạo ra bởi N hình chữ nhật đã cho.

**Test ví dụ:**

Input	Ouput
2	8
3	6
4 28 27 11	
15 20 42 5	
11 24 33 14	
5	
4 28 27 11	
12 11 34 2	
7 26 14 16	
14 16 19 12	
17 28 27 21	

Giới hạn thời gian: 5s

Giới hạn bộ nhớ: 524288 Kb

**BÀI B. BIỂU DIỄN THẬP PHÂN CÓ KHÔNG QUÁ 2 CHỮ SỐ**

Cho một số nguyên dương N, hãy đếm xem có bao nhiêu số nguyên dương trong đoạn  $[1, N]$  có biểu diễn thập phân gồm không quá 2 chữ số.

**Input:**

Một số nguyên dương N duy nhất ( $1 \leq N \leq 10^9$ ).

**Output:**

In ra một số nguyên duy nhất là đáp án của bài toán.

**Test ví dụ:**

Input	Ouput
10	10
105	101
356	175

Giải thích test 2: Có 4 số không thỏa mãn là 102, 103, 104, 105.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

**BÀI C. HOÁN VỊ**

Cho số nguyên N và một hoán vị P[] có độ dài 2N, gồm các phần tử  $\{1, 2, \dots, 2N-1, 2N\}$ . Có 2 loại thao tác:

1. Đổi chỗ các phần tử P[1] vs P[2], P[3] vs P[4], ..., P[2N-1] vs P[2N].
2. Đổi chỗ các phần tử P[1] vs P[N+1], P[2] vs P[N+2], ..., P[N] vs P[2N].

Nhiệm vụ của bạn là hãy sử dụng ít phép biến đổi nhất để đưa hoán vị P[] ban đầu về thành một dãy số tăng dần.

**Input:**

Dòng đầu tiên là số nguyên N ( $1 \leq N \leq 1000$ ).

Dòng tiếp theo gồm  $2N$  số nguyên  $P[i]$  mô tả một dãy hoán vị.

**Output:**

In ra số thao tác ít nhất cần sử dụng. Nếu không tìm được đáp án, hãy in ra -1.

**Test ví dụ:**

Input:	Output:
3 6 3 2 5 4 1	3
2 3 4 2 1	-1
3 1 2 3 4 5 6	0

Giải thích test 1:

Sử dụng thao tác 1: 3 6 5 2 1 4

Sử dụng thao tác 2: 2 1 4 3 6 5

Sử dụng thao tác 1: 1 2 3 4 5 6

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

## BÀI D. MỞ KHOÁ

Cho một chiếc khóa số có  $D$  kí tự, được đánh số là từ 0, 1, ...,  $D-1$ . Mỗi bước, bạn có thể ấn một nút để lựa chọn một chữ số. Sau khi ấn thì không thể bật nút đó lại được nữa. Tuy nhiên, bạn có thể ấn nút RESET để bật tất cả các nút đã chọn về trạng thái ban đầu.



Chiếc khóa có 1 mật khẩu duy nhất, khi tổ hợp các nút được chọn trùng với tổ hợp mật khẩu, chiếc khóa sẽ được mở. Mặc dù bạn không biết mật khẩu chính xác là gì, nhưng trên tờ hướng dẫn sử dụng đã có ghi sẵn  $N$  mật khẩu tiềm năng.

Nhiệm vụ của bạn là hãy tính số bước bấm nút ít nhất có thể, để có thể đảm bảo rằng tất cả các mật khẩu tiềm năng đều đã được thử qua, và chiếc khóa sẽ chắc chắn được mở. Khi ấn nút RESET sẽ được tính là 1 bước.

**Input:**

Dòng đầu tiên gồm 2 số nguyên  $D$  và  $N$  ( $1 \leq D \leq 10$ ,  $1 \leq N \leq 2^D - 1$ ).

$N$  dòng tiếp theo, mỗi dòng gồm  $D$  kí tự, mô tả một mật khẩu. Chỉ số của chuỗi mật khẩu được đánh số từ vị trí 0. Kí tự thứ  $j$  của mật khẩu bằng 1, biểu diễn tổ hợp mật khẩu có chứa chữ số  $j$ .

Input đảm bảo các mật khẩu là phân biệt, và trong mỗi mật khẩu sẽ chứa ít nhất một kí tự 1.

**Output:**

In ra một số nguyên là số lần bấm nút ít nhất để có thể mở khóa.

**Test ví dụ:**

Input	Output
2 2 10 11	2
3 4 001 111 101 011	6

**Giải thích test 1:** Hai mật khẩu tiềm năng là 0 và 01.

Ấn nút 0 trước, rồi ấn nút 1, chắc chắn chiếc khóa sẽ được mở ở bước 1 hoặc 2.

**Giải thích test 2:** Bốn mật khẩu tiềm năng là 2, 012, 02 và 12.

Một giải pháp là 2 0 RESET 1 2 0.

Ấn 2 và 0, xử lý được 2 mật khẩu tiềm năng là 2 và 02.

Ấn 1 2 0, xử lý được 2 mật khẩu tiềm năng là 12 và 012.

Một phương án khác đó là ấn theo thứ tự 1 2 RESET 2 0 1.

*Giới hạn thời gian: 2s*

*Giới hạn bộ nhớ: 524288 Kb*

## BÀI E. NHÂN 3 HOẶC CHIA 2

Cho dãy số  $A[]$  có  $N$  phần tử. Mỗi bước biến đổi, bạn cần thực hiện một trong hai loại thao tác sau lên các phần tử  $A[i]$ :

(1) nhân  $A[i]$  với 3

(2) lấy  $A[i]$  chia cho 2.

Phép chia được thực hiện khi và chỉ khi  $A[i]$  là một số chẵn. Tại mỗi bước, không được phép thực hiện duy nhất thao tác (1) lên cả  $N$  phần tử.

Nhiệm vụ của bạn là hãy xác định xem số lượng bước biến đổi có thể được thực hiện lớn nhất bằng bao nhiêu?

**Input:**

Dòng đầu tiên là số nguyên dương  $N$  ( $1 \leq N \leq 10000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $1 \leq A[i] \leq 10^9$ ).

**Output:**

In ra đáp án là mức năng lượng tìm được.

**Test ví dụ:**

Input	Output
3 5 2 4	3
4 1 3 5 7	0

6	21
242 192 333 169 256 192	

**Giải thích test 1:**

Bước 1: Nhân A[1] với 3, nhân A[2] với 3, chia A[3] cho 2 à {15, 6, 2}

Bước 2: Nhân A[1] với 3, chia A[2] cho 2, nhân A[3] với 2 à {45, 3, 6}

Bước 3: Nhân A[1] với 3, nhân A[2] với 3, chia A[3] cho 2 à {135, 9, 3}

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

**BÀI F. DÃY SỐ CÓ CÁC PHẦN TỬ BẰNG NHAU**

Cho dãy số A[] có N phần tử là một hoán vị của 1, 2, 3, ..., N. Bạn được phép thực hiện phép biến đổi sau: mỗi bước, chọn một đoạn K phần tử liên tiếp, sau đó gán giá trị của cả đoạn này bằng giá trị phần tử nhỏ nhất trong đoạn.

Các bạn hãy tính xem cần sử dụng ít nhất bao nhiêu phép biến đổi để có thể thu được một dãy số mới có tất cả các phần tử đều bằng nhau?

**Input:**

Dòng đầu tiên là số nguyên dương N và K ( $2 \leq K \leq N \leq 100000$ ).

Dòng tiếp theo là N số nguyên mô tả dãy số hoán vị A[].

**Output:**

In ra một số nguyên là số bước đổi ít nhất cần thực hiện.

**Test ví dụ:**

Input	Output
3 3 1 3 2	1
4 3 2 3 1 4	2
8 3 4 2 3 1 5 6 8 7	4

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

**BÀI G. DÃY SỐ LŨY THỪA 2**

Cho dãy số A[] gồm các số nguyên dương là lũy thừa của 2.

Tại mỗi bước, bạn được phép chọn 1 trong 2 thao tác sau:

- Loại bỏ 1 số bất kỳ
- Hoặc cộng 2 số giống nhau và liền kề với nhau trở thành 1 số mới.

Các thao tác trên dừng lại khi chỉ còn 1 số duy nhất.

Hãy tính xem giá trị lớn nhất có thể đạt được là bao nhiêu.

Ví dụ dãy A[] ban đầu là {4,2,2,1,8} thì các thao tác có thể như sau:

- Cộng 2 số 2 liền kề -> {4,4,1,8}
- Cộng 2 số 4 liền kề -> {8,1,8}
- Loại bỏ số 1 -> {8,8}
- Cộng 2 số 8 liền kề -> {16}

Và 16 cũng là giá trị lớn nhất có thể đạt được.

### Input

Dòng đầu ghi số N là số phần tử của dãy A[] ( $1 \leq N \leq 1000$ ).

Dòng thứ 2 ghi N số của dãy A[], trong đó  $1 \leq A[i] \leq 2^{100}$ .

### Output

Ghi ra giá trị lớn nhất có thể đạt được.

### Ví dụ

Input	Output
5 4 2 2 1 8	16

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 524288 Kb

## BÀI H. TẬP CON

Cho dãy số A[] với N số nguyên phân biệt.

Hãy tìm tất cả các số nguyên X sao cho tồn tại 2 tập con không giao nhau trong dãy A[] mà tổng các phần tử trong mỗi tập đó bằng X.

Một tập con của một dãy số được tạo ra bằng cách loại bỏ một số phần tử (không loại bỏ tất cả) trong dãy số ban đầu.

Dãy số ban đầu cũng được coi là tập con của chính nó (tức là không loại bỏ phần tử nào).

### Input:

- Dòng đầu ghi số N là số phần tử của dãy số A[] ( $1 \leq N \leq 18$ ).
- Dòng số 2 ghi N số nguyên phân biệt  $A_1; A_2; \dots; A_N$  ( $1 \leq A_i \leq 30$ ).

### Output:

- Dòng đầu tiên ghi số m là số lượng giá trị X tìm được.
- Dòng thứ 2 lần lượt ghi các giá trị X theo thứ tự từ nhỏ đến lớn.

### Ví dụ:

Input	Output
4 1 3 4 2	3 3 4 5
1 4	0

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

## BÀI I. XOÁ ĐỈNH

Cho đồ thị vô hướng với N đỉnh và M cạnh. Tại mỗi một thời điểm, một đỉnh X được lựa chọn để xóa, đồng thời, tất cả các cạnh kết nối với X cũng bị xóa bỏ.

Nhiệm vụ của bạn là hãy xác định xem đồ thị còn lại có liên thông hay không? (Tức là tất cả các cặp đỉnh đều có thể kết nối được với nhau).

### Input:

Dòng đầu tiên gồm hai số nguyên N và M.

M dòng tiếp theo, mỗi dòng gồm 2 số nguyên u và v mô tả cạnh từ u à v.

N dòng tiếp theo chứa một hoán vị P[] của 1, 2, ..., N là thứ tự các đỉnh sẽ bị xóa.

**Giới hạn:**

$2 \leq N, M \leq 200\,000$ .

**Output:**

In ra N dòng, “YES” nếu đồ thị liên thông, “NO” trong trường hợp ngược lại.

Dòng đầu tiên là kết quả của đồ thị ban đầu. Dòng thứ i+1 thể hiện kết quả của đồ thị còn lại sau khi đỉnh P[i] bị xóa.

**Lưu ý:** Không in ra kết quả cho đỉnh P[N] vì đồ thị sau cùng là rỗng.

**Ví dụ:**

Input	Output
4 3	YES
1 2	NO
2 3	YES
3 4	YES
3	
4	
1	
2	
4 4	YES
1 2	YES
2 3	YES
3 4	YES
4 1	
1	
2	
3	
4	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 524288 Kb

## BÀI J. BẦU CỬ

Khu dân cư ABC tiến hành bầu tổ trưởng dân phố. Có M ứng viên và N cử tri. Người dân trong khu dân cư đã chán ngấy với việc các ứng viên vận động tranh cử, câu kéo phiếu bầu trong các nhiệm kỳ trước nên họ quyết định đặt ra quy định mới như sau:

- Các ứng viên được đánh số từ 1 tới M. Mỗi cử tri sẽ viết ra đúng 1 số thứ tự ứng viên mình muốn chọn và bỏ vào hòm phiếu.
- Người trúng cử là người có số phiếu bầu **nhiều thứ hai**
- Nếu không có người đứng thứ hai thì kết quả bầu cử sẽ bị hủy bỏ
- Nếu có nhiều hơn 1 người cùng có số phiếu nhiều thứ hai thì người nào có số thứ tự nhỏ nhất sẽ được chọn.

Viết chương trình xác định người trúng cử.

**Input**

Dòng đầu ghi hai số N và M ( $1 < M < 10, 5 < N < 500$ ).

Dòng thứ 2 ghi N giá trị trong các phiếu bầu. Các giá trị đảm bảo hợp lệ (tức là từ 1 đến M).

### **Output**

Ghi ra số thứ tự của người trúng cử.

Hoặc nếu không có ai trúng cử thì ghi ra NONE

### **Ví dụ**

<b>Input</b>	<b>Output</b>
10 4 2 3 1 2 3 4 1 2 3 2	3
8 4 1 2 3 4 4 3 2 1	NONE

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 524288 Kb

-----**HẾT**-----