

**Tổng quan về đề thi**

Bài	Tên file bài làm	Tên file dữ liệu	Tên file kết quả	Giới hạn mỗi test	Điểm
1	PUNCH.*	PUNCH.INP	PUNCH.OUT	1 giây - 1024MB	6
2	BRIDGES.*	BRIDGES.INP	BRIDGES.OUT	1 giây - 1024MB	7
3	FFILL.*	FFILL.INP	FFILL.OUT	1 giây - 1024MB	7






Phần mở rộng \* là PAS, C hay CPP tùy theo ngôn ngữ và môi trường lập trình (Free Pascal hay CodeBlock)

Đề có 3 trang.

**Hãy lập chương trình giải các bài toán sau đây**

**Bài 1. BÌA ĐỤC LỖ**

Trong triển lãm lịch sử máy tính, hiện vật được chú ý nhiều nhất là những mảnh bìa đục lỗ. Có  $n$  mảnh bìa đánh số từ 1 tới  $n$ , mỗi mảnh là một hình vuông kích thước  $4 \times 4$  với 16 ô vuông đánh số từ 0 tới 15 theo thứ tự của các ô được chỉ ra như trong hình bên trái

0	1	2	3				
4	5	6	7				
8	9	10	11				
12	13	14	15				

Để lưu dữ liệu bằng những mảnh bìa, mỗi ô có thể đục lỗ (ứng với bit 1) hoặc không (ứng với bit 0). Mỗi mảnh bìa ứng với một chuỗi nhị phân 16 bit biểu diễn một giá trị nguyên trong phạm vi từ 0 tới  $2^{16} - 1$ . Cụ thể là nếu mảnh bìa ứng với chuỗi nhị phân  $x_{15}x_{14} \dots x_1x_0$  trong đó:

$$\begin{cases} x_i = 0, & \text{nếu ô thứ } i \text{ không bị đục lỗ} \\ x_i = 1, & \text{nếu ô thứ } i \text{ bị đục lỗ} \end{cases}$$

Thì mảnh bìa đó sẽ biểu diễn giá trị:

$$\sum_{i=0}^{15} x_i 2^i = x_{15} \cdot 2^{15} + x_{14} \cdot 2^{14} + \dots + x_1 \cdot 2 + x_0$$

Ví dụ như trong hình trên bên phải, mảnh bìa biểu diễn chuỗi nhị phân 0000010000010111 ứng với giá trị:  $2^{10} + 2^4 + 2^2 + 2^1 + 2^0 = 1047$

Các mảnh bìa được xếp khít lên nhau thành một chồng. Để giữ chúng cố định, người ta chọn hai vị trí khác nhau và xuyên hai đinh sắt qua tất cả các ô tại hai vị trí đó của các tấm bìa. Tuy vậy với những ô chưa đục lỗ, việc xuyên thủng các ô đó là rất đáng tiếc vì có thể phá hủy hiện vật.

**Yêu cầu:** Bạn được phép xoay các tấm bìa đi 90 độ, mỗi tấm bìa có thể xoay số lần tùy ý nhưng không được lật mặt. Hãy tìm cách xoay và xác định hai vị trí xuyên đinh sắt sao cho số lần phải xuyên thủng các ô chưa đục lỗ là ít nhất.

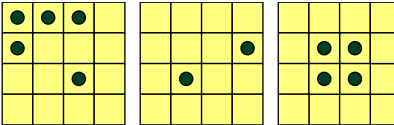
**Dữ liệu:** Vào từ file văn bản PUNCH.INP

- Dòng 1 chứa số nguyên dương  $n \leq 10^6$
- Dòng 2 chứa  $n$  số nguyên dương cách nhau bởi dấu cách, số thứ  $i \in [0; 2^{16} - 1]$  là giá trị số mà mảng bit thứ  $i$  biểu diễn

**Kết quả:** Ghi ra file văn bản PUNCH.OUT một số nguyên duy nhất là số lần phải xuyên thủng các ô chưa đục lỗ theo phương án tìm được

**Ví dụ**

PUNCH . INP	PUNCH . OUT
3 1047 640 1632	1



**Chú ý:** 50% số điểm ứng với các test có  $n \leq 10000$

## Bài 2. CẦU KHỈ

Một trong những trò chơi dân gian được các bạn trường Trần Phú sáng tạo trong dịp kỷ niệm 30 năm thành lập trường là trò đi qua cầu: Có  $N$  viên gạch được xếp thành một hàng dọc đánh số 1, 2, ...,  $N$ . Giữa viên gạch  $i$  và viên gạch  $i + 1$  các bạn nữ đã chuẩn bị một cây cầu mỏng manh bằng các thanh gỗ mỏng và chỉ chịu được  $a_i$  lần đi qua nó. Như vậy có  $N - 1$  cây cầu tất cả. Người đến lượt chơi sẽ bắt đầu xuất phát từ một viên gạch nào đó, đi qua đi lại các cây cầu chừng nào nó còn chưa bị phá hủy; Mỗi lần đi qua một cây cầu, điểm của người chơi sẽ được tăng thêm 1. Tất nhiên quá trình này sẽ dừng khi đến một viên gạch không còn cầu nguyên lành nào để đi tiếp. Hùng được chơi đầu tiên. Hãy tìm số điểm lớn nhất mà cậu ta có thể nhận được.

**Dữ liệu:** Vào từ file văn bản BRIDGES.INP

- Dòng 1: Ghi số nguyên dương  $n$  ( $2 \leq n \leq 10^6$ )
- Dòng 2: Ghi  $n - 1$  số nguyên  $a_1, a_2, \dots, a_{n-1}$  trong đó  $a_i$  là số lần tối đa có thể đi qua cầu nối viên gạch  $i$  với viên gạch  $i + 1$

**Kết quả:** Ghi ra file văn bản BRIDGES.OUT một số nguyên duy nhất là số điểm lớn nhất mà Hùng có thể nhận được.

**Ví dụ:**

BRIDGES . INP	BRIDGES . OUT
5 2 1 2 1	5

Hùng xuất phát từ viên gạch 3  
đi theo hành trình  $3 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2$

Subtask (50% số điểm):  $n \leq 5000$

## Bài 3. TÔ MÀU

Một bức ảnh đen trắng được số hóa dưới dạng một bảng các điểm ảnh kích thước  $m \times n$ . Các hàng điểm được đánh số từ 1 tới  $m$  từ trên xuống dưới và các cột điểm được đánh số từ 1 tới  $n$  từ trái qua phải. Điểm ảnh nằm trên hàng  $i$ , cột  $j$  gọi là điểm  $(i, j)$ . Mỗi điểm ảnh có màu đen (B) hoặc trắng (W)

Hai điểm ảnh được gọi là **thông nhau** nếu chúng cùng màu và đứng cạnh nhau trên cùng hàng hoặc cùng cột. Ta gọi một **miền** là một tập hợp tối đại các điểm cùng màu thỏa mãn: Giữa hai điểm bất kỳ của miền, ta có thể đi từ vị trí điểm này đến vị trí điểm kia qua một số phép di chuyển giữa

hai ô thông nhau. Tính tối đại ở đây có nghĩa là việc bổ sung bất kỳ điểm ảnh nào vào **miền** sẽ làm cho tính chất trên bị vi phạm.

Giáo sư X đang muốn kiểm thử hoạt động cho một chương trình tô màu. Chương trình này có thể thực hiện những lệnh dạng *FloodFill*( $c, i, j$ ): Tô màu  $c$  cho toàn bộ miền chứa điểm  $(i, j)$ .

Bởi sau mỗi lệnh *FloodFill*, số lượng và sự phân bố các miền có thể thay đổi. Với một dãy lệnh *FloodFill*, giáo sư X muốn biết số lượng miền và kích thước miền lớn nhất (tính bằng số điểm ảnh) sau mỗi lệnh.

**Dữ liệu:** Vào từ file văn bản FFILL.INP

- ✧ Dòng 1 chứa hai số nguyên dương  $m, n \leq 1000$  cách nhau bởi dấu cách
- ✧  $m$  dòng tiếp theo, dòng thứ  $i$  chứa  $n$  ký tự  $\in \{B, W\}$  liền nhau. Ký tự thứ  $j$  là màu của điểm  $(i, j)$
- ✧ Dòng tiếp theo chứa số nguyên dương  $q \leq 10^5$  là số lệnh *FloodFill*
- ✧  $q$  dòng tiếp theo, mỗi dòng chứa ký tự  $c \in \{B, W\}$  tiếp theo là hai số nguyên  $i, j$  cách nhau bởi dấu cách là tham số của một lệnh *FloodFill*( $c, i, j$ ) mô tả như trên ( $1 \leq i \leq m; 1 \leq j \leq n$ )

**Kết quả:** Ghi ra file văn bản FFILL.OUT  $q$  dòng, dòng thứ  $p$  chứa hai số cách nhau bởi dấu cách: Số thứ nhất là số miền trên ảnh, số thứ hai là kích thước miền lớn nhất sau lệnh *FloodFill* thứ  $p$ .

**Ví dụ**

FFILL.INP	FFILL.OUT	FFILL.INP	FFILL.OUT
3 3	5 5	2 4	2 7
BWB	1 9	WWBB	2 7
WBW		WBWB	
BWB		2	
2		W 1 3	
W 2 2		W 2 4	
B 2 3			

**Chú ý:** 50% số điểm ứng với các test có  $m, n, q \leq 200$

∞ HẾT ∞