**Bài 1.**

Đề bài: Cho một ma trận M x N gồm các số từ 0~200.

In ra số lượng các số lớn nhất trên ma trận theo chiều dọc, ngang.

Ví dụ:

Ở hàng thứ nhất, số lớn nhất là 10

Hàng thứ 2, số lớn nhất là 11

Hàng thứ 3, số lớn nhất là 9

Hàng thứ 4, số lớn nhất là 8

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 6 | 9 | 10 |
| 7 | 2 | 11 | 5 |
| 2 | 1 | 5 | 9 |
| 0 | 2 | 3 | 8 |

Tiếp tục chọn theo cột, chọn ra được:

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 6 | 9 | 10 |
| 7 | 2 | 11 | 5 |
| 2 | 1 | 5 | 9 |
| 0 | 2 | 3 | 8 |

Vậy nhưng mỗi hàng, cột chỉ được phép tồn tại 1 số, vì vậy số 6,7,9,8 sẽ bị loại bỏ

Những số được chọn cuối cùng:

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 6 | 9 | 10 |
| 7 | 2 | 11 | 5 |
| 2 | 1 | 5 | 9 |
| 0 | 2 | 3 | 8 |

Số lượng các số được chọn là 2.

Nếu trên 1 hàng - cột có 2 số lớn nhất là bằng nhau thì giữ lại cả 2 số.

**TESTCASE**

Dòng đầu tiên là số test case T.

Dòng tiếp theo là số hàng (M) và cột (N)

M dòng tiếp theo là M hàng x N cột của ma trận

**Input**

5

4 4

5 6 9 10

7 2 11 5

2 1 5 9

0 2 3 8

6 6

2 3 8 11 1 4

9 8 2 10 2 1

8 5 4 1 8 2

3 4 5 6 7 8

22 11 7 2 4 9

1 10 9 2 3 8

5 6

14 3 22 28 26 3

14 29 24 1 13 7

7 13 14 19 12 0

26 14 24 27 28 19

19 5 13 23 16 20

5 6

21 7 12 12 8 14

10 2 10 29 20 12

16 24 29 14 9 5

21 1 17 9 5 28

18 7 7 13 21 14

5 6

13 28 0 20 3 22

12 5 9 20 22 0

9 6 16 5 3 4

8 27 15 16 8 9

21 29 0 19 13 16

Ex1.

#1 2

#2 3

#3 3

#4 5

#5 3

**Bài 2**

Cho một bàn cờ NxN.

Trong bàn cờ có **1** quân mã được đánh số là 2.

Trên bàn cờ có các quân cờ được đánh số là 1.

Các ô không có quân cờ nào được đánh số là 0.

In ra xem tất cả các quân mã đó có thể ăn được bao nhiêu quân trên bàn cờ.

Ví dụ:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0x I -2, j -1 | 0 | 1 i – 2, j + 1 | 0 | 0 | 0 | 0 |
| 0xi-1, j -2 | 0 | 0 | 1 | 0x I -1, j + 2 | 1 | 0 | 0 |
| 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| 0xi + 1, j -2 | 0 | 0 | 0 | 0x I + 1, j + 2 | 0 | 0 | 0 |
| 0 | 0xi+2, j -1 | 0 | 1i + 2, j + 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

**In ra sẽ là 2.**

**Testcase**

Dòng thứ nhất là số lượng test case

Dòng thứ 2 là số N là kích thước của mảng

N dòng tiếp theo là các giá trị trong mảng biểu thị quân mã, quân bình thường, và ô trống

6

8 8

1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 1

1 1 1 1 1 1 0 1

1 0 1 2 1 1 1 1

1 0 0 0 1 1 0 1

1 0 1 0 1 1 1 1

1 0 0 1 1 0 1 1

0 0 1 1 0 1 1 1

8 8

1 1 1 0 0 0 1 1

0 0 1 0 1 1 0 1

1 1 1 1 0 1 0 1

0 1 0 2 1 1 0 0

1 1 0 0 1 0 1 1

0 1 0 0 0 1 0 1

1 1 1 0 0 1 1 1

0 1 1 0 0 0 1 1

8 8

0 1 0 1 0 1 1 1

1 0 0 0 1 0 1 1

1 0 0 0 0 1 0 1

0 0 0 0 0 0 1 0

1 1 0 0 0 0 0 0

1 0 0 0 2 1 1 1

0 0 0 1 0 0 0 0

0 1 1 1 0 1 1 1

16 16

1 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0

0 0 0 0 0 0 1 1 0 1 1 0 1 1 0 1

0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 0

1 0 1 0 0 0 1 1 1 1 1 1 0 0 0 1

1 0 1 1 0 0 0 1 2 1 1 1 1 1 1 1

1 0 1 0 0 0 0 1 0 0 1 0 1 1 1 1

1 0 0 1 0 0 1 0 0 0 1 0 0 1 1 1

0 0 0 1 0 1 1 0 0 1 1 0 1 1 1 0

1 1 1 1 0 0 0 0 0 1 0 0 1 0 1 0

1 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1

0 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1

1 0 1 1 0 1 0 1 1 0 1 1 1 1 0 1

0 0 0 0 1 0 0 1 1 1 1 0 0 1 1 0

0 1 1 1 1 0 1 0 1 0 0 0 1 0 1 0

1 0 0 0 1 1 1 0 1 0 0 0 1 0 0 1

1 1 0 1 1 0 0 1 0 1 0 1 0 1 1 1

16 16

0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0

1 0 1 1 1 1 0 0 1 0 0 0 0 1 1 0

0 0 1 1 1 1 1 1 0 1 1 0 1 0 1 1

1 1 0 1 0 0 0 1 2 1 0 0 1 0 1 0

0 1 0 0 0 0 0 1 0 1 1 1 0 1 0 0

0 0 0 0 1 0 0 1 0 1 0 0 1 1 1 1

1 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0

0 0 0 1 0 1 0 1 1 1 1 1 1 1 0 1

1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 1

0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 1

1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 0

1 0 1 0 0 1 0 0 0 0 0 1 1 1 1 0

0 1 0 0 1 0 1 1 0 1 0 1 1 1 0 1

1 0 1 0 1 0 0 1 1 0 1 0 1 1 0 0

1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0

0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1

16 16

0 0 0 0 1 1 1 0 0 0 1 0 1 0 0 0

1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 0

1 0 0 0 0 1 1 1 0 1 1 0 1 0 1 0

0 0 0 1 0 1 1 0 1 1 1 1 0 0 0 0

0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1

0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1

0 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1

1 1 0 1 1 0 0 1 1 1 0 0 0 0 0 1

0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1

0 0 0 1 1 0 1 0 1 0 2 0 1 0 1 0

1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0

1 1 0 1 1 0 0 0 0 0 0 1 0 1 1 0

1 1 0 1 1 1 0 0 0 1 0 0 1 1 1 0

0 0 0 0 0 0 0 1 1 0 1 1 0 1 1 1

0 0 0 0 1 1 1 1 0 1 0 0 1 1 1 0

1 1 1 0 1 1 0 0 1 0 1 1 0 1 0 0

#1 5

#2 5

#3 2

#4 4

#5 5

#6 3

**Bài 3**

Bài này giống bài 2, chỉ khác có nhiều quân mã

Cho một bàn cờ NxN.

Trong bàn cờ có các quân mã được đánh số là 2.

Trên bàn cờ có các quân cờ được đánh số là 1.

Các ô không có quân cờ nào được đánh số là 0.

In ra xem tất cả các quân mã đó có thể ăn được bao nhiêu quân trên bàn cờ.

Ví dụ:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

**Màu đỏ là vị trí của các quân mã**

**Màu vàng là các quân có thể ăn được**

**Các quân mã không ăn được nhau**

**Với testcase này in ra 3 là đáp án đúng**

**Testcase**

Dòng thứ nhất là số lượng test case

Dòng thứ 2 là số N là kích thước của mảng

N dòng tiếp theo là các giá trị trong mảng biểu thị quân mã, quân bình thường, và ô trống

5

8 8  
1 1 0 1 1 0 2 1   
1 0 0 0 1 0 0 1   
0 0 0 0 1 1 0 0   
1 1 0 2 0 0 1 0   
0 0 0 0 1 1 0 0   
0 1 0 1 1 1 1 0   
0 2 0 1 1 0 1 1   
1 0 0 1 0 1 0 1   
  
8 8  
0 0 1 1 1 0 1 1   
0 2 0 1 1 0 0 0   
0 0 1 1 1 2 0 1   
1 1 1 0 1 0 0 0   
0 0 1 1 1 0 1 0   
1 2 1 1 1 1 1 0   
0 1 0 1 1 1 0 1   
0 1 0 0 1 2 1 1   
  
8 8  
1 1 0 0 0 0 1 0   
1 1 0 2 0 0 0 1   
0 0 1 0 0 1 1 1   
0 0 1 1 0 2 0 0   
0 0 1 0 1 1 1 1   
0 2 0 0 0 1 1 0   
1 0 1 0 1 1 1 1   
1 1 1 2 1 1 1 0

15 15

0 0 0 1 1 1 0 0 1 0 0 0 1 2 1

1 1 1 1 0 1 1 1 1 0 1 0 0 0 1

0 1 1 1 2 0 0 1 0 0 0 0 0 0 1

1 1 0 0 1 0 1 0 0 0 1 1 0 0 1

1 0 1 1 0 1 1 1 0 0 0 0 0 0 0

1 1 0 1 1 0 0 2 0 0 1 1 1 1 0

1 1 1 1 0 0 0 0 0 0 1 1 0 0 0

1 0 0 1 1 1 0 0 1 1 1 0 0 1 0

1 1 0 1 0 0 0 0 1 0 1 1 1 1 0

0 1 0 0 0 1 0 1 0 0 1 0 1 0 0

2 0 1 0 0 0 1 0 1 1 1 0 0 1 0

0 0 0 0 1 1 0 1 0 1 1 1 0 1 1

0 1 1 2 1 1 1 0 1 0 1 0 1 1 1

0 0 0 1 1 1 0 0 0 1 1 1 1 1 1

0 1 1 0 1 1 0 0 1 1 2 1 1 1 0

15 15

1 0 1 0 0 1 1 2 1 1 0 0 0 0 0

1 0 1 1 1 0 1 1 1 1 0 0 1 0 0

0 0 1 1 0 0 1 0 1 0 0 1 1 1 0

1 0 1 0 0 0 1 0 0 1 1 1 0 1 0

1 1 0 0 0 1 1 0 0 2 0 0 0 0 1

1 1 1 1 1 1 1 0 1 1 0 0 1 1 1

2 1 0 1 0 1 0 0 0 1 1 0 1 0 0

1 1 0 1 0 1 1 1 0 1 1 1 0 1 1

1 1 1 0 0 1 1 1 1 0 0 0 0 1 1

0 1 1 0 1 1 1 1 1 1 0 0 0 1 1

1 0 1 0 0 0 0 1 0 0 1 1 0 0 0

0 1 0 1 1 0 0 0 0 0 1 1 1 1 1

0 0 1 0 1 0 1 0 1 0 0 1 1 1 1

1 0 1 1 0 1 1 1 0 0 0 0 1 1 0

1 0 0 0 0 1 2 0 0 1 1 0 1 0 0

#1 6

#2 14

#3 9

#4 17

#5 8

**Bài 4**

Tại một số quốc gia, có một số các số được coi là số đẹp hơn các số khác.

Ví dụ tại đất nước A, số 5 và số 7 là số đẹp.

Samsung muốn bán được nhiều sản phẩm nhất tại nước A, nhưng người nước A chỉ mua sản phẩm khi sản phẩm đó được đánh số trong đó có n số đẹp.

Ví dụ:

Các sản phẩm được đánh số từ x - y, (x,y < 100 000) ví dụ trong trường hợp này là từ

0 <= mã sp <= 1000, muốn bán được sản phẩm cần phải chứa ít nhất 2 số đẹp trong đó (là 5 và 7).

Vậy nên các số như 55 , 77, 57, 75, 55x, 77x, 57x, 75x… được coi là những sản phẩm có số đẹp và bán được.

Yêu cầu in ra tổng các số thỏa mãn các yêu cầu trên.

Ví dụ như trên có đáp án là : 55 + 77 + 57 + 75 + 77x + 55x + 57x + 75x + 7x7 + 5x5 + 7x5 + 5x7 + x55 + x57 + x75 + x77 = 4 + 8\*10 + 4\*9 = 120 số.

**Test case:**

Dòng đầu tiên là số lượng test case

3 dòng tiếp theo

Dòng thứ 1là số lượng các số đẹp (n số), số lượng các số đẹp yêu cầu có trong mã sản phẩm ( m số)

Dòng thứ 2 là n số đẹp liên tiếp nhau

Dòng thứ 3 là 2 số x,y. (số chọn nằm trong khoảng x <= số <= y)

**Output:**

In ra số lượng các số thỏa mãn

**Input:**

**Sai truong hop dau**

4

2 3

2 0

15 900

3 2

1 2 3

1 99

3 3

4 8 9

123 9921

3 3

2 7 3

1199 22132

Ex4.

#1 4

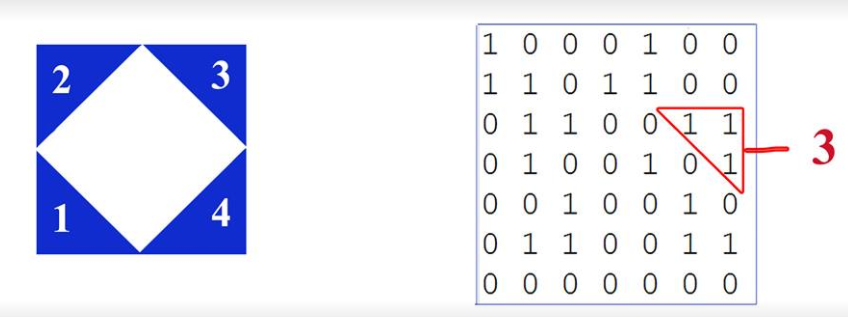
#2 9

#3 792

#4 2149

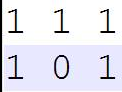
**Bài 5**

**Có 4 loại tam giác đánh số 1,2,3,4 như hình vẽ. Cho ma trận cấp N\*N... vs các phần tử 0 và 1. Y/c: Đếm số lượng các tam giác xuất hiện trong ma trận.**

****

**Lưu ý: Một số có thể dùng nhiều lần để vẽ tam giác.**

**Ví dụ:**

****

**Số 1 ở giữa có thể được dùng 2 lần để vẽ ra 2 tam giác.**

**Đếm số tam giác xuất hiện.**

**Testcase:**

**Dòng đầu tiên là số lượng testcase**

**Dòng tiếp theo là N, kích cỡ của ma trận NxN**

**N dòng tiếp theo là ma trận chứa 0 - 1.**

**5**

**10 10**

**0 0 0 0 0 1 0 1 0 0**

**1 0 1 0 1 1 1 1 0 0**

**0 0 0 0 0 0 0 1 1 0**

**0 1 1 0 1 0 0 1 1 1**

**1 1 1 1 1 1 0 1 0 1**

**1 0 1 1 1 0 0 0 1 1**

**1 1 0 1 0 1 0 1 0 0**

**0 1 0 1 1 1 1 0 1 1**

**1 1 1 0 0 1 1 0 1 1**

**1 0 1 1 0 1 1 1 1 1**

**10 10**

**1 1 0 1 1 0 1 0 1 0**

**1 1 0 1 0 1 0 1 1 0**

**0 0 1 1 1 0 1 0 0 0**

**1 1 0 0 0 1 1 1 0 0**

**1 1 0 0 0 1 1 0 1 0**

**0 0 0 0 1 0 0 1 1 0**

**0 0 0 0 1 0 0 0 1 0**

**0 0 0 0 0 0 1 1 1 1**

**0 1 1 0 0 1 1 1 1 1**

**1 1 1 1 1 0 0 0 0 0**

**10 10**

**1 1 1 0 1 1 0 1 1 0**

**0 0 1 1 0 1 1 1 0 0**

**1 1 1 0 0 1 0 0 1 1**

**0 1 1 1 0 0 1 0 1 0**

**0 0 1 1 1 0 0 1 1 0**

**0 0 0 1 1 1 0 0 0 1**

**0 1 1 1 1 0 0 1 1 1**

**0 0 0 1 0 0 0 0 1 1**

**0 1 1 0 1 1 1 1 1 0**

**0 1 0 1 1 0 0 1 1 1**

**10 10**

**0 1 1 1 1 0 1 1 0 0**

**1 1 0 0 0 1 0 0 1 1**

**1 0 0 0 0 0 1 1 0 0**

**1 1 0 0 0 0 1 0 1 0**

**1 0 0 0 0 0 1 0 0 0**

**0 1 1 1 1 0 1 0 0 1**

**0 0 1 1 0 0 1 0 0 1**

**1 0 1 1 1 1 1 0 1 1**

**0 0 0 0 1 0 0 0 0 0**

**1 0 0 1 0 1 0 1 1 1**

**10 10**

**0 0 0 1 0 1 0 1 1 1**

**0 0 1 1 0 1 1 1 0 1**

**1 1 0 1 1 1 0 1 0 1**

**0 0 0 1 0 0 1 1 0 0**

**0 0 0 1 1 1 1 0 1 0**

**1 0 1 1 1 1 1 1 0 1**

**0 0 1 1 1 0 0 1 1 0**

**1 1 1 0 1 0 1 0 0 1**

**0 1 0 1 1 1 1 0 0 0**

**0 1 0 1 0 1 0 0 1 1**

**Ex5.**

**#1 63**

**#2 42**

**#3 54**

**#4 21**

**#5 51**

**Bài 6**

**cho ma trận kích thước n \* n gồm các ký tự 0 và 1 ,ví dụ n = 3**

**TH1,**

**100**

**100**

**111**

**Th2,**

**111**

**010**

**010**

**TH3,**

**101**

**101**

**010**

**TH4**

**101**

**111**

**101**

**3 trường hợp trên lần lượt là**

**+ 1 chữ L,0 chữ T,0 chữ U, 0 chữ H**

**+0 chữ L ,1 chữ T,0 chữ U, 0 chữ H**

**+0 chữ L,0 chữ T,1 chữ U, 0 chữ H**

**+0 chữ L,0 chữ T,0 chữ U, 1 chữ H**

**N <= 100 các chữ cách nhau và luôn có nghĩa**

**Mỗi chữ nằm trong 1 block 3x3 và được bao bọc tối thiểu bởi 1 dãy số 0**

**Các chữ có thể xoay 90-180-270 độ**

**Test case:**

**Dòng đầu số test case, dòng 2 kích thước ma trận vuông N x N**

**N dòng tiếp theo là các ký tự trong mảng**

**3**

**10 10**

**1 0 0 0 0 0 1 1 0 0**

**1 0 0 0 0 0 0 0 0 0**

**1 1 1 0 0 0 1 0 1 0**

**0 0 0 0 0 0 1 0 1 0**

**1 1 1 0 0 0 0 1 0 0**

**0 1 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 1 0 0**

**0 0 1 0 0 0 0 1 0 0**

**0 0 1 0 0 0 0 1 1 1**

**0 1 1 1 0 0 0 0 0 0**

**10 10**

**1 1 1 0 1 0 0 0 0 0**

**0 0 0 0 1 0 0 0 0 0**

**1 1 1 0 1 1 1 0 0 0**

**0 1 0 0 0 0 0 0 0 0**

**1 1 1 0 1 1 1 0 0 0**

**0 0 0 0 0 0 1 0 0 0**

**0 0 0 0 0 0 1 0 0 0**

**0 0 0 1 0 0 0 0 1 0**

**0 0 1 0 1 0 0 0 1 0**

**0 0 1 0 1 0 0 0 1 0**

**10 10**

**0 0 0 1 0 0 0 0 0 0**

**0 1 0 0 0 0 0 0 0 0**

**1 0 1 0 1 1 0 0 0 0**

**1 0 1 0 0 0 1 0 0 0**

**0 0 0 0 1 1 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 1 0 0 0 0 1 1 1**

**1 1 1 0 0 0 0 1 0 0**

**0 0 1 0 0 0 0 1 0 0**

**Output**

**In ra 4 số theo thứ tự là số lượng chữ H, T, L, U đếm được**

**Ex6.**

**#1 0 1 2 1**

**#2 1 0 2 1**

**#3 0 1 1 2**

**Bài 7**

**Người ta muốn tìm một đường trượt băng trên một khu vực núi.**

**N: là số điểm**

**H: là chiều cao so với mặt đất tại điểm đó**

**F: là lực để di chuyển. Biết khi F < 0 thì k thể trượt được. F = 0 sẽ trượt sang trái hoặc phải nếu bên đó thấp hơn điểm đang đứng.**

**Khi di chuyển từ cao (H1)-> thấp(H2) : F = 1 đơn vị lực x ( H1 - H2).**

**Khi di chuyển từ thấp(H1)-> cao (H2): F = -3 đơn vị lực x ( H1 - H2).**

**Tìm đoạn lớn nhất có thể làm sân trượt tuyết.( khoảng cách 2 đỉnh )**

**ví dụ:**

**Số đầu là test case**

**N: 10**

**H(1->10):** [**5 20 18 15 10 7 6 1**](https://www.facebook.com/) **10 0**

**Đáp án: 6 ( đi từ 20 -> 1)**

**Đi từ 20 xuống 1 được 19 lực, đi từ 1 lên 10 mất 9\*3 = 27 lực, ko đi được.**

**H(1->10):** [**5 20 15 10**](https://www.facebook.com/) **5** [**0 2 1 6**](https://www.facebook.com/) **7**

**Đáp án: 4 Từ 20 -> 0 hoặc từ 7->0**

**Đi từ 20 xuống 0 được 20 lực, đi từ 0 lên 2 mất 6 lực, còn 14**

**Đi từ 2 xuống 1 được thêm 1 là 15**

**Từ 1 lên 6 mất 15 lực là còn 0 lực.**

**Vật đang ở 6 chỗ cao, sau khi lên được sẽ lăn lại xuống 1 là được 5 lực**

**Lăn tiếp từ 1 lên 2 mất 3 lực, còn 2 lực**

**Lăn từ 2 xuống 0 được 2 lực, là 4 lực**

**Lăn từ 0 lên 5 mất 15 lực, ko đủ nên vật dừng lại.**

**Đường trượt tuyết được chỉ được tính đến lúc dừng lại, vì vậy chỉ đc tính từ 20-0**

**Test:**

**5**

**10**

**5 9 4 1 0 3 2 5 2 0**

**20**

**9 8 1 2 8 2 0 8 1 0 1 4 8 5 4 1 0 1 4 9**

**9**

**9 1 0 11 2 9 20 1 8**

**20**

**23 7 18 20 2 8 22 15 19 24 1 3 13 19 6 0 0 13 11 11**

**20**

**7 11 8 9 14 7 19 23 13 4 2 2 3 5 23 18 11 17 20 12**

**Ex7. Output chuẩn**

**#1 5**

**#2 4**

**#3 2**

**#4 3**

**#5 3**

**Bài 8**

Giả sử có một trò chơi xếp hình như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Các block màu vàng là các khối hình 2x1

Sau đó, các khối hình rơi xuống dưới thành hình dạng như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

3 1 2 1 2 2

Khối 2x1 bôi đỏ sau khi rơi xuống 1 đầu bị khối bên dưới giữ lại nên đầu còn lại ko rơi được xuống như hình.

Nhiệm vụ: In ra chiều cao cao nhất ở mỗi hàng sau khi tất cả các block rơi xuống, như trong ví dụ trên là: 3 1 2 1 2 2

Biết rằng các khối luôn tách biệt nhau và ko dính vào nhau để loại trừ trường hợp không biết block 2x1 là dọc hay ngang như sau đây:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Testcase**

**Dòng 1 là số lượng test case**

**Dòng tiếp theo là số hàng và cột ( bài này luôn là 10 hàng - 10 cột)**

**10 dòng tiếp theo là input, 0 là vị trí trống, 1 là vị trí có gạch**

**Các block luôn là 2x1**

**5**

**10 10**

**1 0 0 0 1 1 0 0 0 0**

**1 0 0 0 0 0 0 0 0 0**

**0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 0 0 0 0 0 0**

**0 0 0 1 0 0 1 1 0 0**

**0 0 0 1 0 0 0 0 0 0**

**0 0 0 0 0 0 0 1 1 0**

**0 1 0 1 1 0 0 0 0 1**

**0 1 0 0 0 0 0 0 0 1**

**0 0 0 0 0 0 0 0 0 0**

**10 10**

**0 0 1 0 0 1 1 0 0 0**

**0 0 1 0 0 0 0 0 0 1**

**0 0 0 0 1 1 0 0 0 1**

**0 1 0 0 0 0 0 0 0 0**

**0 1 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 1**

**0 1 1 0 1 1 0 0 0 1**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 1 0 0 0 0 0 0**

**0 0 0 1 0 1 1 0 0 0**

**10 10**

**0 0 0 0 0 0 1 0 0 0**

**0 1 1 0 0 0 1 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**1 0 0 0 0 0 1 0 0 0**

**1 0 1 1 0 0 1 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 1 0 0 0 0 0 0 0 0**

**0 1 0 0 0 0 1 0 0 0**

**0 0 0 1 1 0 1 0 0 0**

**0 0 0 0 0 0 0 0 1 1**

**10 10**

**0 0 0 0 1 1 0 0 0 1**

**0 0 0 0 0 0 0 0 0 1**

**1 1 0 0 0 0 0 0 0 0**

**0 0 0 1 0 1 0 0 0 0**

**0 0 0 1 0 1 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 1 1 0 0 1 0 0**

**0 0 0 0 0 0 0 1 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 1 1 0 1 1 0 0 0 0**

**10 10**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 1 0 0 0 0**

**0 0 0 0 0 1 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**1 1 0 0 0 1 0 0 0 0**

**0 0 0 0 0 1 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 1 1 0 0 0**

**0 0 1 1 0 0 0 0 1 1**

**Ex8.**

**#1 5 3 4 4 2 2 2 2 1 2**

**#2 0 3 3 2 3 4 4 0 0 4**

**#3 2 3 3 2 1 0 6 0 1 1**

**#4 2 2 1 4 4 4 0 2 0 2**

**#5 1 1 1 1 0 5 1 0 1 1**

**Bài 9**

**Có n quốc gia cạnh nhau và sắp xếp như một mảng một chiều, mỗi quốc gia có quân đội bảo vệ lãnh thổ của mình, số lượng của binh sĩ trong quân đội nằm trong mỗi ô như hình dưới:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **20** | **8** | **16** | **9** | **35** | **7** |

**A B C D E F**

**Giả sử đất nước A muốn mở rộng lãnh thổ nên tấn công theo hướng bên phải sang nước B.**

**Trong trận chiến giữa các đất nước, khi đất nước bị tấn công cảm thấy binh lực bên mình chỉ <= ½ binh lực đội quân tấn công, đất nước đó sẽ đầu hàng và toàn bộ binh lực bị nước tấn công thâu tóm. Còn nếu binh lực > 1/2 , 2 nước sẽ chiến đấu với nhau theo tỉ lệ thiệt hại 1-1,đất nước nào còn quân sẽ chiến thắng.**

**Giả sử khi tấn công chỉ được tấn công theo 1 hướng từ trái qua phải hoặc từ phải qua trái mà không được quay lại. Đất nước nào tấn cống sẽ tấn công cho đến khi hết binh. Đất nước được chọn khi tấn công kể cả khi binh lực ít hơn cũng vẫn tấn công theo tỉ lệ 1-1 mà không sáp nhập vào đất nước khác.**

**In ra số lượng quân tối đa mà một đất nước có thể đánh bại, số quân đánh bại tính cả số quân thu phục về nước mình.**

**Ví dụ đất nước A tấn công sang bên phải, mỗi bước sẽ là :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **20** | **8** | **16** | **9** | **35** | **7** |

**A B C D E F**

**20/2 > 8, sẽ thu phục được 8 lính để thành 28 quân**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **28** | **12** | **3** | **35** | **7** |

**A B C D E F**

**28/2 < 16, sau khi đánh nước C sẽ chỉ còn 12 quân**

**12/2 < 9, sau khi đánh nước D sẽ chỉ còn 3 quân**

**Và tiếp tục đánh nước E sẽ hết quân.**

**Vậy tổng số quân mà nước A đánh bại là: 8 + 16+ 9 + 3 = 36 quân.**

**Tiếp tục tính toán cho đất nước B,C,D,E,F và in ra số lớn nhất mà 1 đất nước đánh bại được.**

**Testcase:**

**Dòng đầu là số lượng test**

**Mỗi test trên 2 dòng**

**Dòng 1 là số lượng đất nước (N)**

**Dòng tiếp theo là N số biểu thị quân mỗi nước**

**5**

**10**

**689 767 390 36 938 916 165 459 503 508**

**10**

**591 330 154 9 94 189 653 259 419 485**

**10**

**220 454 763 840 909 384 108 739 880 570**

**20**

**47 144 103 10 177 1 199 30 113 4 60 134 126 134 181 88 51 71 23 174**

**20**

**6 24 148 48 60 141 89 125 177 111 195 114 147 116 147 99 95 91 24 54**

Ex9.

#1 689 839 462 36 1790 1635 165 789 503 508

#2 609 1222 172 9 112 207 1367 259 419 485

#3 220 454 763 840 1893 600 108 1723 880 570

#4 47 164 123 10 197 1 1189 30 173 4 68 322 126 134 407 88 51 117 23 1696

#5 [6 24](https://www.facebook.com/) 364 48 60 286 89 125 177 111 195 114 147 116 147 99 95 91 24 102

**Bài 10**

**Nhân dịp đến quê hương Bắc Giang tham gia thi chọn học sinh giỏi Trại hè Hùng Vương, Hà đến thăm trang trại trồng vải thiều của bác Thuấn. Trang trại gồm n cây thẳng hàng và cách đều nhau (khoảng cách từ gốc cây i đến gốc cây i+1 là 3 mét với mọi i = 1, 2, 3, ..., n-1).**

**Năm nay là năm được mùa nên cây nào cũng sai quả, cây thứ i có số lượng quả vải là ai quả. Bác Thuấn rất lo lắng vì tuổi già rồi làm sao có thể hái được hết số vải thiều này, Hà đang nghĩ cách giúp bác Thuấn thì đột nhiên có một ông tiên hiện ra trước mắt Hà và nói: “Ta cho cháu một chiếc túi thần, cháu hãy chọn một chỗ đứng ở một gốc cây trong n cây rồi hô to câu thần chú [*Vải ơi chui vào đây*], khi đó tất cả quả ở các cây có khoảng cách tính từ gốc đến chỗ cháu đứng không vượt quá H (mét) sẽ từ từ rời khỏi cành và bay vào túi thần này.”**

### Yêu cầu

**Hãy cho biết Hà có thể thu hoạch giúp bác Thuấn được tối đa bao nhiêu quả vải nếu lời nói của ông tiên là hiện thực.**

### Input

**Đầu vào có dạng: Số testcase đầu tiên**

* **Dòng đầu chứa hai số nguyên dương n, H (n ≤ 100; H ≤ 109)**
* **Dòng thứ hai chứa n số nguyên a1 , a2 , . . ., an trong đó ai (ai ≤ 109 ; i = 1, 2, 3, . . ., n) là số lượng quả của cây vải thứ i.**

**Hai số liên tiếp trên cùng dòng được ghi cách nhau bởi dấu cách.**

### Output

**Gồm một số nguyên duy nhất là tổng số lượng quả vải nhiều nhất mà Hà có thể thu hoạch giúp bác Thuấn.**

**6**

**5 6**

**5 53 7 13 78**

**5 7**

**72 21 49 61 63**

**5 11**

**1 53 35 57 2**

**15 14**

**1 57 97 34 8 97 70 51 12 69 56 95 80 38 79**

**15 22**

**8 67 4 7 50 9 37 37 67 67 65 40 69 68 88**

**15 25**

**20 78 63 12 98 53 7 41 49 51 65 13 75 61 14**

**Ex10.**

**#1 156**

**#2 266**

**#3 148**

**#4 568**

**#5 683**

**#6 700**

**Bài 11**

**Cho một bộ test cast gồm:**

**Dòng đầu tiên là số lượng testcase, mỗi testcase gồm 2 dòng**

**Dòng đầu tiên gồm 2 số là n và m trong đó n là số lượng các số, m là tổng yêu cầu của n số trên.**

**Dòng tiếp theo là n số đó.**

**Biết rằng trong n số là các số từ 1 tới 9 và không lặp lại nhau.**

**Sẽ có 2 số ở mỗi dòng chưa được gán giá trị được gọi là x và y và được biểu thị là 0.**

**Yêu cầu in ra số cách có thể điền 2 số đó. Nếu không có in ra 0**

**Ví dụ:**

**5 20**

**2 3 0 0 8**

**Ta thấy có 2 số cần điền gọi là x và y. 2 + 3 + x + y + 8 = 20 -> x + y = 7**

**Vì các số là từ 1 tới 9 và không trùng nhau, vậy x và y có thể là:**

**x = 6 , y = 1 hoặc x = 1, y = 6. (không thể là 2 5 do đã có số 2)**

**Tổng cộng có 2 cách có thể điền x và y.**

**Testcase**

**5**

**4 15**

**1 2 0 0**

**5 14**

**1 2 0 9 0**

**5 22**

**1 9 8 0 0**

**5 25**

**2 8 4 0 0**

**5 30**

**9 8 7 0 0**

**Output**

**#1 6**

**#2 0**

**#3 0**

**#4 2**

**#5 4**

**Bài 12.**

Cho một hệ thống đèn gồm 0 là đèn tắt và 1 là đèn sáng.

Khi đặt một đèn chiếu laser vào mỗi cột của ma trận thì đèn tắt sẽ sáng và đèn sáng sẽ tắt.

Hỏi sau M lần đặt đèn chiếu laser thì số lượng đèn sáng tối đa có thể hiển thị sau cùng là bao nhiêu.

Ví dụ

5

0 1 1 1 1

0 0 1 0 1

0 0 0 0 0

0 0 0 1 0

Đặt 1 lần vào cột số 1 cả 4 đèn đều sáng

Đặt 1 lần vào cột số 2 thì 3 đèn sáng và 1 đèn thành tắt

3 lần tiếp theo đặt vào cột 3,4,5 thì số lượng đèn tắt, sáng không thay đổi.

Tối đa thu được là 13 đèn sáng.

Ma trận kích thước 10 x 10 cho mọi test

Cho test case gồm

Dòng đầu n là số lượng test

N bộ tiếp theo là số test

Tiếp theo là M là số lần đặt đèn chiếu laser.

X hàng tiếp theo là ma trận biểu thị đèn tắt - sáng

5

21

10 10

1 0 0 0 0 0 1 0 1 0

1 0 0 0 1 0 0 0 1 0

0 1 1 0 0 1 1 1 1 0

1 0 0 1 1 1 1 0 1 0

0 1 1 0 0 1 0 0 1 1

1 1 0 0 0 0 1 1 1 0

0 0 1 0 0 0 0 1 0 1

1 1 0 1 1 0 0 0 1 1

0 0 0 1 0 0 0 1 1 0

0 0 1 0 0 1 0 0 1 0

33

10 10

0 0 0 1 1 1 0 0 1 1

0 0 0 0 1 0 1 1 0 0

1 0 1 1 0 0 1 0 0 0

0 1 1 1 1 0 0 1 0 0

1 1 0 1 1 0 1 1 1 1

1 1 1 1 0 1 0 1 1 1

0 1 1 0 1 0 0 1 1 0

1 1 1 0 1 0 0 0 1 1

1 0 0 1 0 0 1 1 1 0

1 0 1 1 1 0 0 0 0 1

25

10 10

0 0 1 0 1 0 1 0 1 0

0 1 1 0 0 1 0 1 0 1

0 0 1 1 0 0 0 0 1 0

0 1 1 1 0 1 0 1 1 1

0 0 0 0 0 1 1 0 0 1

0 1 1 1 1 0 1 1 0 0

0 0 1 1 1 1 0 0 0 0

0 1 0 0 1 0 1 0 1 0

0 0 1 0 1 0 1 1 0 1

0 0 1 0 1 1 1 1 0 0

30

10 10

1 1 0 1 1 0 1 0 0 0

1 1 1 1 1 1 0 1 1 1

0 1 0 0 0 1 1 1 1 1

1 1 1 1 0 1 0 0 0 0

1 1 1 0 0 1 0 1 1 0

1 1 0 0 0 0 0 0 1 0

0 0 0 0 0 0 1 1 0 0

0 0 1 0 0 1 1 1 0 1

0 1 0 1 0 1 0 1 0 0

0 1 1 0 0 0 1 0 0 1 d

44

10 10

0 1 1 0 0 0 1 0 0 0

1 0 1 1 0 0 0 0 1 1

0 0 0 0 0 0 1 0 1 0

1 1 0 1 0 1 1 0 1 0

1 0 0 1 1 1 1 0 0 0

1 1 0 1 1 0 1 1 0 1

0 1 0 0 0 1 1 0 0 0

1 0 0 1 0 1 1 1 0 1

0 0 0 1 0 0 1 1 1 0

1 0 0 0 1 0 1 0 0 0

Output:

#1 65

#2 62

#3 64

#4 61

#5 66

**Bài 13. Cho một ma trận NxM, yêu cầu: Tìm các hình vuông xuất hiện trong ma trận, các hình vuông có kích cỡ từ 2x2 tới 5x5, biết rằng các hình vuông được bao quanh bởi ít nhất một dãy số 0.**

**Ví dụ:**

**0 0 0 0**

**0 1 1 0 Đây là một hình vuông**

**0 1 1 0**

**0 0 0 0**

**1 0 0 0**

**0 1 1 0 Không được tính là một hình vuông**

**0 1 1 0**

**0 0 0 0**

**Trường hợp hình vuông ở góc thì không cần bao ở cạnh đó nữa.**

**Ví dụ:**

**1 1 0**

**1 1 0**

**0 0 0**

**Cũng là 1 hình vuông**

**Input:**

**5**

**10 10**

**1 1 0 0 0 1 1 1 1 1**

**1 1 0 1 1 1 0 1 0 0**

**0 0 0 1 0 1 1 0 1 0**

**0 1 1 0 1 1 1 0 0 0**

**1 1 1 1 1 0 0 1 1 1**

**0 0 0 1 1 0 0 0 0 0**

**1 1 1 1 1 0 1 1 1 1**

**0 1 1 0 1 0 1 1 1 1**

**1 1 1 0 1 0 1 1 1 1**

**1 1 0 0 1 0 1 1 1 1**

**10 10**

**0 1 1 0 0 0 1 1 1 1**

**0 1 1 0 1 0 1 1 1 1**

**0 0 0 0 1 0 1 1 1 1**

**1 0 0 1 1 0 1 1 1 1**

**0 0 1 1 1 0 0 0 0 0**

**0 0 0 1 0 1 1 0 0 0**

**0 1 0 0 1 1 0 0 0 0**

**1 0 1 1 1 0 0 1 1 0**

**1 1 1 1 1 0 0 1 1 0**

**1 0 1 0 0 0 0 0 0 0**

**10 10**

**1 1 0 1 1 0 0 1 1 1**

**1 1 0 1 1 0 0 0 0 0**

**0 0 0 0 0 0 0 1 1 0**

**0 0 0 0 1 1 1 1 1 0**

**1 1 1 0 0 1 1 1 0 0**

**1 1 1 0 0 1 0 0 1 0**

**1 1 1 0 0 1 0 1 0 1**

**0 0 0 0 0 1 1 0 0 0**

**1 1 0 1 0 0 1 0 1 1**

**1 1 0 1 1 1 1 0 1 1**

**10 10**

**1 1 1 1 1 0 1 1 0 0**

**1 1 1 1 1 0 0 0 0 0**

**1 1 1 1 1 0 1 1 1 1**

**1 1 1 1 1 0 0 1 1 0**

**1 1 1 1 1 0 0 1 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 1 0 0 1 0 1 1 0 1**

**0 0 0 1 0 1 1 1 1 1**

**1 1 0 0 1 1 0 0 0 0**

**1 1 0 1 0 0 1 0 1 1**

**10 10**

**1 0 0 0 1 1 0 1 0 1**

**1 0 0 1 0 1 0 1 0 0**

**0 1 1 1 0 0 1 1 1 0**

**0 0 0 1 1 1 1 1 1 0**

**1 0 0 0 0 1 1 0 0 1**

**1 1 1 1 1 1 1 0 0 1**

**1 0 1 1 0 1 1 0 1 0**

**0 0 0 1 0 0 1 0 1 1**

**0 0 1 1 1 1 1 1 0 1**

**0 0 0 0 0 1 0 1 0 1**

**# Output**

**2**

**3**

**5**

**2**

**0**

**Bài 14.**

**Slot machine là trò chơi giống như hình dưới:  
**

Khi 3 ô số 7 trùng nhau sẽ đạt được phần thưởng lớn nhất.

Giả sử: Khi 3 ô số 7 trùng nhau sẽ được 100 điểm.

Các ô khác khi có 3 ô trùng nhau sẽ được 10\*X điểm với X là số đó.

Nếu không có 3 ô nào trùng nhau số điểm thu được sẽ bằng tổng của 3 ô đó. Các số dao động từ 1 tới 9.

Ví dụ:

1 ô slot machine sau có điểm là:

1 1 2

7 7 7

3 3 3

Tổng điểm = 1 + 1 + 2 + 100 + 3\*10 =134 điểm.

Cho một máy slot machine kích thước 7x3, với 3 hàng slot machine hiển thị lên màn hình là 3 hàng ở giữa. 2 hàng trên và 2 hàng dưới sẽ là những số có thể được kéo vào slot machine nếu ta kéo lên trên hoặc xuống dưới.

Giả sử chúng ta chỉ có thể chọn 1 hàng duy nhất để kéo lên hoặc kéo xuống với tối đa là 2 ô.

Ví dụ:

7 3

8 6 6

7 9 5

**5 5 9**

**3 5 8**

**2 7 8**

2 6 9

5 1 3

3 hàng slot machine là 3 hàng in đậm. Nếu ta chọn kéo cột đầu tiên lên trên 1 hàng thì 3 hàng slot machine sẽ trở thành như sau:

**3 5 9**

**2 5 8**

**2 7 8**

Hỏi số điểm lớn nhất thu được là bao nhiêu.

Testcase:

5

7 3

5 4 7

5 1 9

2 6 5

8 1 2

4 6 9

1 9 5

7 5 1

7 3

8 8 4

3 7 8

1 8 8

7 6 7

6 3 4

8 5 2

4 2 8

7 3

6 7 3

5 5 5

7 7 8

2 2 3

7 5 3

2 3 7

2 7 3

7 3

5 4 4

2 8 2

9 2 7

5 6 1

9 6 4

6 9 4

3 6 8

7 3

3 7 2

3 3 5

1 4 5

4 9 6

1 5 7

1 2 9

2 2 1

OUTPUT:

50

135

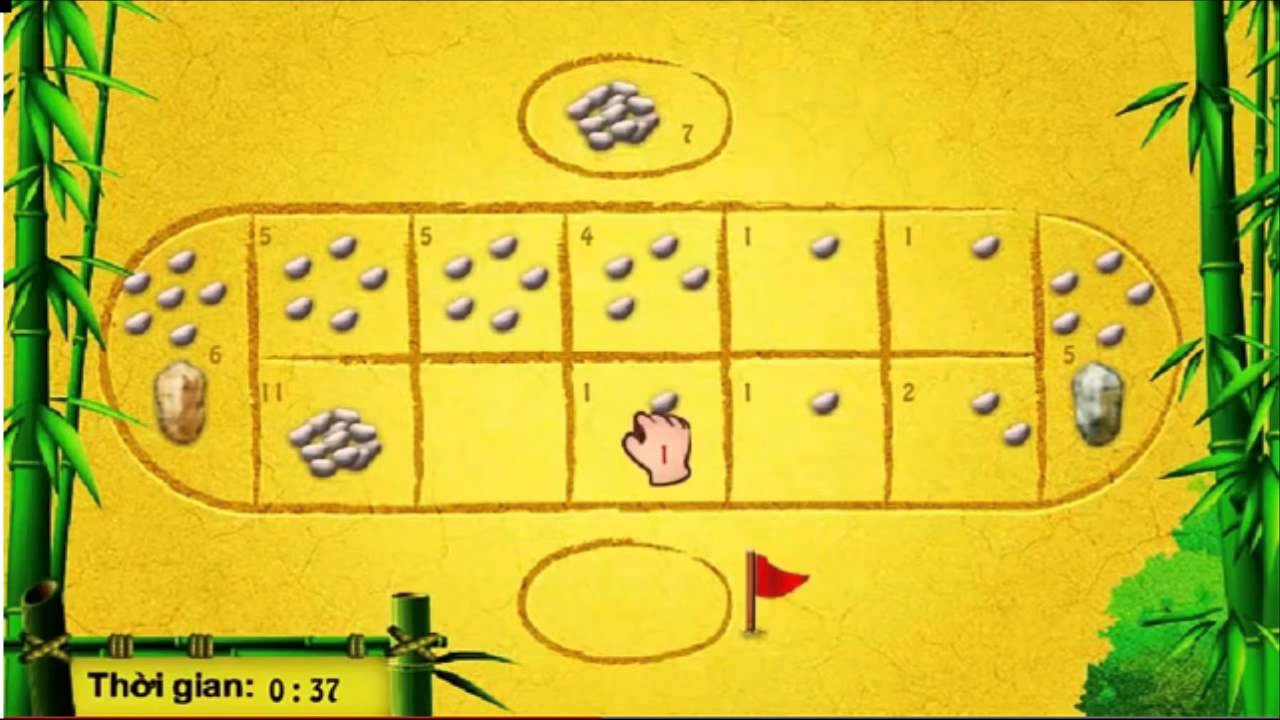
49

56

46

**Bài 15**

Cho một trò chơi ô ăn quan như hình sau:



Tìm số quân tối đa có thể ăn được khi chọn 1 ô đi theo bên trái hoặc bên phải. Biết các ô trên bàn được hiển thị trên output theo thứ tự như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 1 | 2 | 3 | 4 | 5 | 6 |
| 11 | 10 | 9 | 8 | 7 |

Biết rằng nếu đi đến số 5 mà hết quân không đi tiếp được sang số 6, trong khi số 6 vẫn còn quân thì dừng lại, số quân ăn đc là 0.

Tương tự với bên số 12.

Input:

Dòng đầu là số lượng test case N

N dòng tiếp theo là số lượng các quân ở trong ô ăn quan, tổng cộng 12 ô. Không có quân nào thì là 0.

5

12

0 5 2 0 5 5 6 4 0 2 3 4

12

5 4 3 5 6 0 0 5 3 5 0 4

12

5 3 0 4 3 6 5 6 0 3 5 2

12

5 0 0 0 6 0 9 1 0 10 4 2

12

7 3 0 10 5 7 6 4 9 0 8 4

Output:

5

11

7

15

10

**Bài 16**

Trò chơi reversi chess là trò chơi như hình bên dưới:  


Nếu ta đặt ô trắng hoặc đen mà có 1 ô trắng or đen khác chặn đầu còn lại theo chiều ngang, dọc, chéo, thì tất cả các ô ở giữa sẽ biến thành màu của mình.

Cho 1 là đen, 2 là trắng.

Ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | y |  |
| 1 | 0 | 0 | x |
| 1 |  | 0 |  |
| 1 | 1 |  |  |

Nếu ta đặt số 1 vào vị trí x, bàn cờ sẽ trở thành:

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 |  |  |
| 1 | 1 | 1 | 1 |
| 1 |  | 1 |  |
| 1 | 1 |  |  |

Nếu đặt ô đen vào vị trí y thì chỉ ăn đc thêm 1 ô

Số ô đen nhiều nhất là : 8

Input:  
N là số lượng testcase.

Tiếp theo là 2 số x,y là hàng và cột.

X dòng tiếp theo là giá trị ma trận.

Những ô chưa có cờ được thể hiện là -1.

Hỏi ta đặt quân cờ như thế nào sao cho in ra số quân tối đa có thể mà cờ đen hoặc cờ trắng đạt được. Số nào lớn hơn thì in ra. Chỉ đánh thêm 1 quân cờ đen hay trắng tùy ý.

10

10 10

-1 -1 0 1 -1 -1 0 0 1 0

0 0 0 -1 -1 1 0 1 0 -1

0 1 0 1 1 1 -1 0 -1 -1

0 1 -1 1 1 -1 0 1 0 -1

1 1 -1 1 -1 1 1 -1 0 0

0 0 -1 1 0 -1 1 -1 1 1

0 0 -1 0 1 -1 0 0 0 0

-1 -1 0 0 0 0 1 1 0 0

-1 0 1 -1 1 1 -1 -1 0 -1

1 -1 0 -1 -1 0 -1 -1 -1 -1

10 10

0 0 1 1 -1 0 -1 -1 -1 1

-1 1 1 1 1 1 0 -1 0 0

1 1 0 -1 -1 1 1 1 -1 0

1 -1 1 -1 -1 0 1 0 1 1

-1 -1 0 -1 -1 0 -1 0 -1 -1

1 0 0 1 0 0 0 0 -1 1

1 -1 1 0 -1 1 1 -1 0 1

1 1 0 0 0 0 -1 1 -1 1

0 -1 1 0 1 -1 -1 1 0 0

1 -1 -1 0 -1 1 -1 -1 1 0

10 10

-1 0 -1 -1 0 0 0 -1 -1 -1

1 0 -1 0 0 1 0 0 1 0

1 0 1 0 0 -1 1 0 -1 1

0 1 -1 1 0 0 1 1 0 1

1 1 1 0 1 0 1 -1 0 0

0 -1 0 0 0 -1 0 -1 0 1

0 -1 0 -1 -1 1 0 0 0 1

1 -1 0 -1 1 -1 1 -1 0 0

-1 0 0 1 -1 -1 1 1 -1 0

-1 0 1 0 -1 0 -1 1 -1 -1

10 10

1 0 -1 1 -1 -1 1 1 1 1

-1 -1 0 0 0 0 1 1 1 0

0 -1 0 1 -1 0 1 1 0 0

1 1 0 1 0 -1 -1 0 0 -1

0 0 1 1 1 1 0 1 1 0

-1 0 1 0 -1 -1 0 1 0 -1

-1 1 -1 -1 -1 0 1 1 1 1

1 -1 -1 1 -1 -1 1 1 -1 0

0 1 -1 0 0 1 1 0 0 0

-1 -1 1 1 -1 0 1 1 -1 1

10 10

1 0 1 1 1 1 -1 1 0 1

-1 -1 0 -1 1 -1 -1 1 -1 0

0 0 -1 1 -1 1 1 0 0 0

1 0 -1 1 0 0 -1 0 1 1

0 -1 0 -1 0 0 -1 1 -1 0

-1 -1 1 -1 0 -1 -1 -1 0 1

1 -1 1 0 0 0 1 -1 0 1

0 -1 1 -1 -1 0 -1 1 -1 1

1 1 0 0 -1 0 0 -1 0 0

0 -1 0 -1 1 1 0 -1 -1 1

10 10

-1 -1 0 -1 0 1 -1 0 0 0

-1 0 -1 0 -1 -1 0 -1 0 -1

1 1 -1 1 -1 0 -1 0 1 -1

1 -1 -1 -1 -1 0 0 1 1 0

-1 0 1 -1 1 0 -1 1 1 -1

1 0 -1 0 1 -1 -1 0 0 1

1 -1 1 -1 1 -1 1 0 -1 1

-1 0 0 1 -1 0 1 1 1 0

1 0 0 1 -1 -1 1 1 1 0

-1 1 1 -1 1 1 0 0 0 0

10 10

1 -1 1 0 -1 -1 1 1 0 1

0 -1 -1 -1 0 1 1 -1 1 0

1 0 -1 1 1 0 -1 -1 -1 0

0 -1 0 1 1 1 0 0 -1 1

-1 0 -1 1 0 1 1 0 0 0

-1 -1 1 1 0 0 0 0 0 0

1 -1 -1 1 0 -1 1 1 -1 1

-1 1 -1 0 1 0 0 -1 1 1

-1 1 1 0 1 1 1 0 1 1

1 0 1 -1 1 0 1 0 1 -1

10 10

-1 -1 0 1 -1 0 -1 1 0 0

0 1 0 -1 -1 1 -1 -1 1 0

-1 0 0 0 0 -1 -1 1 0 -1

1 0 1 1 1 -1 1 1 -1 0

0 -1 0 0 1 -1 1 1 1 -1

0 -1 0 1 1 1 0 0 1 1

0 0 -1 0 -1 0 0 -1 -1 0

-1 1 -1 -1 -1 -1 -1 0 1 -1

-1 0 -1 0 -1 1 -1 0 1 0

-1 -1 1 -1 0 0 0 1 -1 1

10 10

0 1 1 0 0 1 1 1 0 1

-1 1 1 -1 -1 1 -1 -1 1 0

1 0 -1 1 -1 0 1 -1 1 -1

-1 1 1 0 0 0 0 0 -1 -1

-1 -1 -1 1 1 1 0 1 1 0

-1 -1 1 0 1 1 -1 0 0 1

-1 -1 1 -1 0 1 -1 1 -1 -1

-1 0 0 1 -1 -1 -1 0 1 0

-1 -1 1 1 1 -1 -1 0 1 0

1 -1 1 1 0 -1 0 -1 1 1

10 10

-1 0 -1 0 0 1 0 -1 0 0

1 -1 1 -1 -1 -1 -1 -1 0 -1

0 -1 -1 -1 0 -1 0 -1 -1 -1

1 1 1 -1 0 1 1 1 -1 1

1 1 1 0 0 1 -1 0 1 1

1 -1 -1 -1 -1 -1 1 1 -1 1

0 0 0 1 1 0 0 1 1 0

0 -1 1 0 1 -1 1 -1 1 0

1 0 1 0 0 -1 0 -1 -1 1

1 0 0 -1 -1 -1 0 -1 1 -1

Output :

44

42

47

46

42

39

48

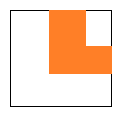
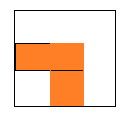
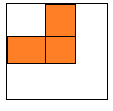
43

45

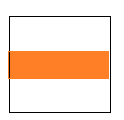
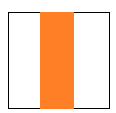
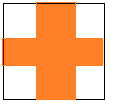
38

**Bài 17.**

Cho một hệ thống ống nước gồm các ống như hình:



1 2 3 4

5 6 7

Ví dụ:

Nếu nối ống 3 nằm cạnh ống 1 nước sẽ thông

Ống 4 nằm cạnh ống 1 nước sẽ thông,

Ống 7 nối với ống nào cũng được

Cho 1 hệ thống các ống nước với các số từ 1 - 7 là mỗi loại ống nước, số 0 là vị trí trống.

Ví dụ hệ thống ống nước dưới đây tương đương với:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |  |  |  |  |  |
| 0 | 5 | 0 | 0 |  |  |  |  |  |
| 3 | 5 | 4 | 0 |  |  |  |  |  |
| 4 | 1 | 0 | 0 |  |  |  |  |  |
| 0 | 0 | 0 | 0 |  |  |  |  |  |
| Yêu cầu: In ra hàng - cột của ống nước bị hỏng, không thông được với ống nào. Trong trường hợp trên là ống được bôi màu xanh, ở hàng 3, cột 3. ( hàng cột tính từ 1). Biết rằng trong mỗi test chỉ có một ống nước hỏng. | | | | | | | | |
|

**Input:**

**5**

**7 8**

**0 0 0 0 0 0 0 0**

**5 0 0 3 0 0 0 0**

**4 6 7 2 0 0 0 0**

**0 0 4 1 0 0 0 0**

**0 0 0 0 0 0 0 0**

**0 0 0 0 3 1 0 0**

**0 0 0 0 4 0 0 0**

**7 8**

**0 0 0 0 3 1 0 0**

**0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0**

**0 0 0 1 0 0 0 0**

**0 0 0 5 0 3 2 0**

**0 0 0 7 0 4 7 0**

**0 0 0 5 0 0 5 0**

**7 8**

**0 0 0 4 2 0 0 0**

**0 3 1 0 5 0 0 0**

**0 7 0 0 7 0 0 0**

**0 5 0 0 6 0 0 0**

**0 4 2 0 0 0 0 0**

**0 0 1 0 0 0 0 0**

**0 0 0 0 0 0 0 0**

**10 10**

**0 0 0 0 0 0 0 0 3 1**

**0 0 0 0 0 0 4 6 7 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 3 2 0 0 0 0 0**

**0 0 0 5 5 0 0 0 0 0**

**0 0 0 7 4 5 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**2 0 0 0 0 0 0 0 0 0**

**1 0 0 0 0 0 3 2 0 0**

**10 10**

**0 2 0 0 0 0 0 0 0 0**

**3 7 0 0 0 0 0 0 0 0**

**0 5 0 0 0 0 0 0 3 1**

**0 4 2 0 0 0 0 6 7 6**

**0 6 1 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 3 6 0 0 0 0 0 0 0**

**2 7 0 0 0 0 0 0 5 0**

**1 5 0 0 0 0 0 0 0 0**

**Output:**

**2 4**

**4 4**

**4 5**

**7 6**

**9 9**

**Bài 18. Cờ ca rô**

**Cho một bàn cờ ca rô kích thước N x M, có 2 người chơi kí hiệu là 1 và 2. Ô số 0 là ô chưa có đánh nước nào.**

**Xác định xem có người chơi nào có thể thắng cuộc nếu họ có cơ hội đánh tiếp 1 nước cờ nữa.**

**Nếu có người thắng cuộc in ra 1, không có ai thắng in ra 0.**

**Điều kiện thắng cuộc là có 5 quân cờ liên tiếp theo hàng ngang, thẳng, chéo, nhưng bị chặn 2 đầu bởi quân địch thì không thắng.**

**Input**

**10 10**

**0 0 0 0 2 0 2 1 1 2**

**1 1 2 2 0 0 0 0 1 1**

**0 0 0 2 0 2 2 2 1 2**

**1 2 2 0 2 0 2 1 2 0**

**1 2 2 2 0 1 0 2 2 1**

**1 1 1 2 2 1 1 0 0 0**

**2 0 1 2 0 2 2 2 1 0**

**1 0 2 0 1 2 2 0 2 0**

**0 0 2 1 2 0 2 0 2 0**

**0 1 0 0 2 0 0 0 1 2**

**10 10**

**2 2 0 1 0 2 2 1 0 2**

**1 1 0 2 1 1 2 2 0 1**

**0 2 2 2 2 1 2 1 0 2**

**0 2 1 2 2 1 0 0 0 1**

**2 0 0 2 0 1 2 2 0 0**

**0 2 0 2 0 0 0 2 1 2**

**2 1 2 0 0 1 2 0 2 1**

**0 0 0 0 0 2 2 1 0 1**

**2 1 0 2 2 0 0 0 1 2**

**1 1 2 1 1 0 2 2 2 1**

**10 10**

**0 1 0 1 2 0 1 1 1 2**

**0 1 1 2 1 2 2 1 0 2**

**2 1 1 0 1 0 0 2 0 0**

**2 2 1 0 1 2 2 2 2 1**

**0 0 0 1 2 0 1 0 2 1**

**0 0 0 0 1 2 2 2 2 0**

**0 1 0 1 2 2 1 1 1 2**

**2 2 0 2 1 2 0 2 2 1**

**1 0 0 1 0 0 2 2 1 1**

**0 1 0 1 2 0 1 2 1 0**

**10 10**

**0 0 1 1 1 2 1 2 2 2**

**0 2 2 0 2 0 1 1 2 2**

**2 1 2 0 1 2 2 0 1 0**

**2 1 1 0 2 2 1 2 2 2**

**1 2 2 2 2 1 1 0 2 0**

**2 1 2 2 0 0 1 0 1 0**

**1 0 1 1 0 0 1 1 2 2**

**1 2 2 0 2 1 2 1 2 2**

**2 1 0 0 0 0 1 1 0 1**

**1 2 2 2 1 2 0 2 1 1**

**10 10**

**0 1 1 2 1 1 2 1 2 1**

**2 1 2 2 0 1 0 2 2 0**

**1 0 1 0 2 2 0 2 0 1**

**0 2 0 0 2 1 0 1 2 2**

**1 1 2 0 1 1 2 2 0 2**

**1 1 2 2 0 0 2 2 2 0**

**2 0 1 2 0 0 2 1 0 0**

**1 2 1 2 1 0 1 0 2 0**

**1 1 2 1 0 2 0 0 0 1**

**2 2 2 0 0 1 0 1 0 2**

**Output.**

**1**

**1**

**1**

**0**

**0**

**Bài 19**

**Tìm mã QR code xuất hiện trong ma trận.**

**QR code có tỉ lệ là 1:1:3:1:1**

**Như hình:**

**1 1 1 1 1 1 1**

**1 0 0 0 0 0 1**

**1 0 1 1 1 0 1**

**1 0 1 1 1 0 1**

**1 0 1 1 1 0 1**

**1 0 0 0 0 0 1**

**1 1 1 1 1 1 1**

**Đây là mã QR viền dày 1. Tương tự với mã viền dày 2, dày 3 độ dày mỗi lớp tăng gấp 2 gấp 3.**

**Tương tự như bài đếm hình vuông, các mã QR code phải được bao quanh bởi ít nhất 1 hàng số 0.**

**In ra nếu test case đó tồn tại QR code.**

**Input:**

**10**

**10 10**

**1 1 1 1 1 1 1 0 1 1**

**1 0 0 0 0 0 1 0 1 1**

**1 0 1 1 1 0 1 0 1 1**

**1 0 1 1 1 0 1 0 1 1**

**1 0 1 1 1 0 1 0 1 1**

**1 0 0 0 0 0 1 0 1 1**

**1 1 1 1 1 1 1 0 1 1**

**0 0 0 0 0 0 0 0 1 1**

**0 1 0 0 0 1 0 0 0 0**

**0 0 0 0 0 0 1 1 1 0**

**10 10**

**1 1 1 1 1 1 1 0 1 1**

**1 0 0 0 0 0 1 0 1 1**

**1 0 1 1 1 0 1 1 1 1**

**1 0 1 1 1 0 1 0 1 1**

**1 0 1 1 1 0 1 0 1 1**

**1 0 0 0 0 0 1 0 1 1**

**1 1 1 1 1 1 1 0 1 1**

**0 0 0 0 0 0 0 0 1 1**

**0 1 0 0 0 1 0 0 0 0**

**0 0 0 0 0 0 1 1 1 0**

**10 10**

**0 1 1 1 1 1 1 0 0 1**

**0 0 0 1 1 1 1 1 0 0**

**0 1 1 1 1 1 1 1 0 0**

**0 1 0 0 0 0 0 1 0 0**

**0 1 0 1 1 1 0 1 0 0**

**0 1 0 1 1 1 0 1 0 0**

**0 1 0 1 1 1 0 1 0 0**

**0 1 0 0 0 0 0 1 0 0**

**0 1 1 1 1 1 1 1 0 0**

**0 0 0 0 0 0 0 0 0 0**

**10 10**

**0 1 1 1 1 1 1 0 0 1**

**0 0 0 0 0 0 0 0 0 0**

**0 1 1 1 1 1 1 1 0 0**

**0 1 0 0 0 0 0 1 0 0**

**0 1 0 1 1 1 0 1 0 0**

**0 1 0 1 1 1 0 1 0 0**

**0 1 0 1 1 1 0 1 0 0**

**0 1 0 0 0 0 0 1 0 0**

**0 1 1 1 1 1 1 1 0 0**

**0 0 0 0 0 0 0 0 0 1**

**10 10**

**0 0 0 1 0 0 0 1 1 1**

**1 1 1 0 0 1 0 0 1 1**

**0 1 0 0 1 1 0 0 1 1**

**0 1 0 0 0 1 1 1 1 0**

**1 0 1 1 0 0 0 0 1 1**

**1 0 0 0 1 0 1 0 0 0**

**0 0 1 1 0 0 0 1 1 1**

**1 1 1 0 1 0 0 0 0 1**

**0 1 0 1 1 1 0 0 1 0**

**0 1 0 1 0 1 0 1 1 0**

**20 20**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 1**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 1 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0**

**1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0**

**1 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1**

**1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0**

**1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0**

**0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 1**

**0 0 1 1 1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 0**

**20 20**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 1 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 1 0 0 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 0**

**1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0**

**1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**1 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1**

**1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0**

**1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0**

**0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 1**

**0 0 1 1 1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 0**

**20 20**

**0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 1 1 0 1**

**0 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 1 0 0**

**0 1 0 1 0 0 1 0 0 0 1 1 0 1 1 1 0 0 0 1**

**0 1 0 0 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 0**

**1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 0**

**0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 1 0 1 0 1**

**0 0 0 0 1 0 0 1 0 0 1 1 1 0 0 0 0 1 0 0**

**1 0 0 1 1 0 1 0 1 1 1 0 0 1 1 0 1 0 0 1**

**1 1 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0**

**0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 1 1 0**

**0 1 0 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 0**

**1 1 1 1 1 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0**

**0 1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0**

**0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0**

**0 0 0 1 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0**

**1 0 1 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 1**

**0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0**

**1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0**

**0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 0 1**

**0 0 0 0 1 1 1 0 0 0 1 1 1 0 1 0 1 0 0 1**

**20 20**

**1 1 1 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 1**

**0 0 1 1 0 1 0 1 1 1 0 0 1 1 1 1 0 0 0 1**

**1 1 1 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 0**

**0 0 0 1 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1**

**1 1 1 1 1 0 1 1 1 0 1 0 0 0 1 1 0 1 1 1**

**1 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 1 1 1 1**

**1 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1**

**1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 0**

**1 1 1 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 0 1**

**0 0 1 1 0 0 1 1 0 1 1 1 0 1 0 1 0 0 0 0**

**0 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0**

**0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 1**

**1 0 0 1 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1**

**1 1 1 1 1 1 0 1 0 0 1 1 0 0 0 1 1 1 0 0**

**1 1 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 1 0 0**

**1 0 1 1 1 1 0 1 0 1 0 1 0 0 1 1 1 0 0 1**

**1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 1 0 1**

**0 1 1 1 0 1 0 1 0 1 1 0 0 0 0 1 1 0 0 1**

**0 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0**

**0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0 1**

**22 21**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

**1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1**

**1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1**

**1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1**

**1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1**

**1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1**

**1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

**Output:**

**#1 Yes**

**#2 No**

**#3 No**

**#4 Yes**

**#5 No**

**#6 No**

**#7 Yes**

**#8 No**

**#9 No**

**#10 Yes**

**Bài 20.**

**Ô ăn quan 2**

Luật chơi tương tự như bài ô ăn quan 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 1 | 2 | 3 | 4 | 5 | 6 |
| 11 | 10 | 9 | 8 | 7 |

Lần này sẽ có 2 người chơi.

Người thứ nhất sẽ chơi bên các ô 1 2 3 4 5

Người thứ hai chơi bên còn lại

Người thứ nhất rất hiếu thắng nên sẽ chọn ô nào để đi có thể ăn được nhiều quân nhất trong mỗi lượt đi.

Người thứ hai thì chơi theo cảm xúc, nếu trường hợp nào có số quân ăn được lớn nhất mà là số lẻ người này sẽ không thích mà quay ra chọn ô nào hiện đang có nhiều quan nhất để đi và đi sang bên phải.

Hỏi sau M vòng số quân của mỗi người ăn được là bao nhiêu, in ra theo thứ tự người thứ nhất - ng thứ 2.

Biết người thứ nhất luôn đi trước. Nếu số lượt đi là lẻ thì người này sẽ đi được nhiều hơn 1 lần so với ng thứ 2.

Input:

Dòng đầu là số lượng test case N bao gồm:

Dòng tiếp theo là tổng số lượt đi của cả 2 người.

Dòng tiếp theo là số lượng các quân ở trong ô ăn quan, tổng cộng 12 ô. Không có quân nào thì là 0.

5

6

14 3 15 7 9 15 14 15 10 0 3 8

4

6 14 13 7 8 11 2 15 0 13 7 13

5

0 0 15 5 3 15 2 7 7 0 6 6

7

10 7 13 11 4 3 13 5 15 13 9 11

8

4 6 1 6 14 12 11 11 6 13 4 9