**[DP]. Bài 1. Xâu con chung dài nhất**

Cho 2 xâu kí tự S và T, hãy tìm xâu con chung dài nhất của S và T. Các kí tự của xâu con không nhất thiết phải liền kề nhau.

**Input Format**

Dòng đầu tiên là xâu S; Dòng thứ 2 là xâu T;

**Constraints**

S và T chỉ gồm các chữ cái in hoa và có độ dài không quá 1000

**Output Format**

In ra độ dài xâu con chung dài nhất của S và T

**Sample Input 0**

ZHFTDFHF

TFISHROV

**Sample Output 0**

3

**[DP]. Bài 2. Dãy con tăng dài nhất (LIS)**

Cho mảng số nguyên A[] gồm N phần tử, hãy tìm dãy con (không nhất thiết các phần tử phải liên tiếp) tăng chặt dài nhất của mảng A.

**Input Format**

Dòng đầu tiên là N; Dòng thứ 2 gồm N phần tử của mảng A;

**Constraints**

1<=N<=1000; 1<=A[i]<=1000;

**Output Format**

In ra độ dài của dãy con tăng dài nhất

**Sample Input 0**

14

128 104 53 876 660 961 754 775 290 231 224 915 392 994

**Sample Output 0**

6

**[DP]. Bài 3. Số bước ít nhất**

Cho mảng A[] gồm N số nguyên. Nhiệm vụ của bạn là sắp xếp lại mảng số với số lượng bước là ít nhất. Tại mỗi bước, bạn chỉ được phép chèn phần tử bất kỳ của mảng vào vị trí bất kỳ trong mảng. Ví dụ A[] = {2, 3, 5, 1, 4, 7, 6 }sẽ cho ta số phép chèn ít nhất là 3 bằng cách lấy số 1 chèn trước số 2, lấy số 4 chèn trước số 5, lấy số 6 chèn trước số 7 ta nhận được mảng được sắp.

**Input Format**

Dòng đầu tiên là N; Dòng thứ 2 gồm N phần tử của mảng A;

**Constraints**

1<=N<=1000; 1<=A[i]<=1000;

**Output Format**

Đưa ra kết quả trên 1 dòng.

**Sample Input 0**

13

143 340 571 845 211 228 274 443 666 594 491 814 24

**Sample Output 0**

6

**[DP]. Bài 4. Dãy con có tổng bằng S**

Cho mảng số nguyên A[] gồm N phần tử và số nguyên dương S, nhiệm vụ của bạn hãy xác định xem có thể tạo ra một tập con các phần tử trong mảng có tổng bằng S hay không? Chú ý mỗi phần tử trong mảng chỉ được sử dụng một lần.

**Input Format**

Dòng đầu tiên gồm 2 số N và S; Dòng thứ 2 gồm N số của mảng A;

**Constraints**

1<=N<=200; 1<=S<=50000; 1<=A[i]<=500;

**Output Format**

In ra 1 nếu có tập con của A có tổng bằng S, ngược lại in ra 0

**Sample Input 0**

8 92

69 16 82 170 31 24 45 112

**Sample Output 0**

1

**[DP]. Bài 5. Staircase**

Một chiếc cầu thang có N bậc. Mỗi bước, bạn được phép bước lên trên tối đa K bước. Hỏi có tất cả bao nhiêu cách bước để đi hết cầu thang? (Tổng số bước đúng bằng N).

**Input Format**

Dòng duy nhất chứa 2 số nguyên N và K

**Constraints**

1<=N<=100000; 1<=K<=100;

**Output Format**

In ra đáp án tìm được trên một dòng theo modulo 10^9+7.

**Sample Input 0**

7 3

**Sample Output 0**

44

**Sample Input 1**

4 2

**Sample Output 1**

5

**[DP]. Bài 6. Frog SPOJ**

Một con ếch có thể nhảy 1, 2, 3 bước để có thể lên đến một đỉnh cần đến. Hãy đếm số các cách con ếch có thể nhảy đến đỉnh.

**Input Format**

Số nguyên dương N mô tả số bước con ếch cần di chuyển để nhảy tới đỉnh

**Constraints**

1<=N<=40

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

5

**Sample Output 0**

13

**Sample Input 1**

10

**Sample Output 1**

274

**[DP]. Bài 7. Xem phim SPOJ**

John có một đàn bò. Một ngày đẹp trời, anh ta quyết định mua xe tải với khả năng chở được C kg (1000 ≤ C ≤ 25000) để đưa những con bò đi xem phim. Cho số con bò là N (20 ≤ N ≤ 100) và khối lượng w[i] của từng con (đều nhỏ hơn C), hãy cho biết khối lượng bò lớn nhất mà John có thể đưa đi xem phim là bao nhiêu.

**Input Format**

Dòng 1: 2 số nguyên C và N cách nhau bởi dấu cách; Dòng 2: Ghi lần lượt các số nguyên: w[i]

**Constraints**

1000 ≤ C ≤ 25000; 20 ≤ N ≤ 100; w[i] <= C;

**Output Format**

Một số nguyên là tổng khối lượng bò lớn nhất mà John có thể mang đi xem phim.

**Sample Input 0**

259 5

81 58 42 33 61

**Sample Output 0**

242

**[DP]. Bài 8. Bài toán cái túi**

Một tên trộm có 1 cái túi có thể mang các đố vật với trọng lượng tối đa là V. Hiện tại tên trộm muốn lựa chọn các đồ vật trong N đồ vật để ăn trộm, mỗi đồ vật có trọng lượng là w[i] và giá trị là v[i]. Hãy xác định tổng giá trị lớn nhất của các đồ vật mà tên trộm này lựa chọn sao cho trọng lượng của chúng không vượt quá V.

**Input Format**

Dòng đầu ghi 2 số N và V. Dòng tiếp theo ghi N số của mảng w. Sau đó là một dòng ghi N số của mảng v.

**Constraints**

V<=1000; N≤1000; 1<=w[i], c[i]<=500;

**Output Format**

In ra giá trị lớn nhất có thể đạt được.

**Sample Input 0**

6 22

39 44 4 59 91 70

47 26 92 33 6 69

**Sample Output 0**

92

**Sample Input 1**

7 22

10 3 22 50 83 16 41

99 97 54 19 50 70 48

**Sample Output 1**

196

**[DP]. Bài 9. Số nguyên lớn**

Cho hai số nguyên lớn N và M có không quá 1000 chữ số. Người ta muốn tính xem liệu có thể lấy ra nhiều nhất bao nhiêu chữ số trong N (không cần liên tiếp) và giữ nguyên thứ tự của nó để tạo ra một số X sao cho ta cũng có thể tìm thấy X trong số M theo cách tương tự.

**Input Format**

Dòng thứ nhất ghi số N, dòng thứ 2 ghi số M

**Constraints**

1<=len(N), len(M) <= 1000

**Output Format**

In ra số chữ số nhiều nhất có thể của X

**Sample Input 0**

82619136359

5572555993152891122

**Sample Output 0**

5

**[DP]. Bài 10. Tổng không liền kề**

Cho mảng A[] gồm N phần tử, nhiệm vụ của bạn là tính tổng lớn nhất của dãy con trong mảng với một điều kiện đó là trong dãy con này không được có 2 phần tử nằm liền kề nhau.

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 là A[i];

**Constraints**

1<=N<=10^6; 1<=A[i]<=1000;

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

7

314 514 148 451 896 589 296

**Sample Output 0**

1706

**[DP]. Bài 11. Xâu con chung dài nhất của 3 xâu**

Cho ba xâu ký tự X, Y, Z. Nhiệm vụ của bạn là tìm độ dài dãy con chung dài nhất có mặt trong cả ba xâu.

**Input Format**

3 dòng lần lượt chứa X, Y, Z;

**Constraints**

1<=len(X), len(Y), len(Z) <= 100;

**Output Format**

In ra độ dài của xâu con chung dài nhất của 3 xâu

**Sample Input 0**

AGLEHHGE

GLGHLALB

DDLBLEHGD

**Sample Output 0**

2

**[DP]. Bài 12. Tổng lớn nhất của dãy con tăng dần**

Cho dãy số A[] gồm N số. Nhiệm vụ của bạn là tìm tổng lớn nhất của dãy con được sắp theo thứ tự tăng dần của dãy A[]. Ví dụ với dãy A[] = {1, 101, 2, 3, 100, 4, 5} ta có kết quả là 106 = 1 + 2 + 3 + 100. Với dãy A[] = {10, 7, 5} ta có kết quả là 10. Với dãy A[] = {1, 2, 3, 5} ta có kết quả là 11.

**Input Format**

Dòng đầu tiên đưa vào N là số phần tử của dãy A[]; Dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống.

**Constraints**

1≤N≤1000; 0≤A[i]≤1000.

**Output Format**

Đưa ra kết quả của bài toán trên 1 dòng

**Sample Input 0**

8

2 12 3 15 3 16 11 4

**Sample Output 0**

45

**[DP]. Bài 13. Dãy số Bitonic**

Một dãy số được gọi là Bi-tonic nếu nó được chia thành hai dãy đầu tăng dần và dãy tiếp theo giảm dần. Nhiệm vụ của bạn là tìm tổng lớn nhất dãy con Bi-tonic của dãy số A[]. Ví dụ với dãy A[] = {1, 15, 51, 45, 33, 100, 12, 18, 9} ta có kết quả là 194 tương ứng với dãy Bi-tonic {1, 15, 51, 100, 18, 9}.

**Input Format**

Dòng đầu tiên đưa vào N là số phần tử của dãy A[]; Dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống

**Constraints**

1<=N<=100; 1<=A[i]<=100;

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

8

7 8 8 19 3 6 2 15

**Sample Output 0**

49

**[DP]. Bài 14. Xúc xắc**

Tèo hiện đang học sinh lớp 10A1, vốn là một học sinh được đánh giá rất cao về khả năng toán học nên Tèo thường được các bạn trong lớp nhờ giải các bài toán trên tạp chí toán học và tuổi trẻ. Hôm nay Tí

là bạn thân của Tèo gặp một bài toán trên tạp chí và đã nhờ Tèo giải hộ, Tèo dù rất giỏi toán và biết cách làm bài này nhưng bài toán này kết quả quá lớn nên Tèo phải nhờ tới bạn lập trình thuật toán để tìm ra lời giải giúp Tèo. Bài toán có yêu cầu rất đơn giản, cho ngẫu nhiên số N, hỏi rằng nếu được tung một con xúc xắc không giới hạn số lần và lấy tổng số chấm trên mặt xúc xắc cộng lại với nhau, hỏi có bao nhiêu cách tung để tổng số chấm trong mọi lần tung xúc xắc bằng N. Bạn hãy giúp Tèo giải quyết bài toán này nhé, vì số cách tung có thể quá lớn, nên bạn hãy chia dư kết quả với modulo 10^9 + 7

**Input Format**

Dòng duy nhất chứa số nguyên dương N

**Constraints**

1<=N<=10^6;

**Output Format**

Kết quả của bài toán

**Sample Input 0**

7

**Sample Output 0**

63

**Sample Input 1**

3

**Sample Output 1**

4

**[DP]. Bài 16. Coin 2**

Hãy xem xét một hệ thống tiền tệ của ngân hàng XYZ bao gồm n đồng xu. Mỗi đồng xu có một giá trị nguyên dương. Nhiệm vụ của bạn là tính số cách riêng biệt mà bạn có thể tạo ra số tiền x bằng cách sử dụng số xu có sẵn. Ví dụ: nếu số xu là {2,3,5} và tổng mong muốn là 9, có 8 cách: 2 + 2 + 5; 2 + 5 + 2; 5 + 2 + 2; 3 + 3 + 3; 2 + 2 + 2 + 3; 2 + 2 + 3 + 2; 2 + 3 + 2 + 2; 3 + 2 + 2 + 2;

**Input Format**

Dòng nhập đầu tiên có hai số nguyên n và x: số xu và số tiền mong muốn. Dòng thứ hai có n số nguyên phân biệt c1, c2,…, cn: giá trị của mỗi đồng xu.

**Constraints**

1≤n≤100; 1≤x≤10^6; 1≤ci≤10^6;

**Output Format**

In ra kết quả lấy dư với 10^9 + 7

**Sample Input 0**

3 9

2 3 5

**Sample Output 0**

8

**[DP]. Bài 17. Coin 3**

Hãy xem xét một hệ thống tiền tệ của ngân hàng ABC bao gồm n đồng xu. Mỗi đồng xu có một giá trị nguyên dương. Nhiệm vụ của bạn là tính số cách riêng biệt không xét đến thứ tự mà bạn có thể tạo ra số tiền x bằng cách sử dụng số tiền có sẵn. Ví dụ: nếu số xu là {2,3,5} và tổng mong muốn là 9, có 3 cách: 2 + 2 + 5; 3 + 3 + 3; 2 + 2 + 2 + 3;

**Input Format**

Dòng nhập đầu tiên có hai số nguyên n và x: số xu và số tiền mong muốn. Dòng thứ hai có n số nguyên phân biệt c1, c2,…, cn: giá trị của mỗi đồng xu.

**Constraints**

1≤n≤100; 1≤x≤10^6; 1≤ci≤10^6;

**Output Format**

In ra kết quả lấy dư với 10^9 + 7

**Sample Input 0**

3 9

2 3 5

**Sample Output 0**

3

**[DP]. Bài 18. Maximum Path Sum**

Cho bảng A[] kích thước N x M (N hàng, M cột). Bạn được phép đi xuống dưới, đi sang phải và đi xuống ô chéo dưới về bên phải. Khi đi qua ô (i, j), điểm nhận được bằng A[i][j]. Hãy tìm đường đi từ ô (1, 1) tới ô (N, M) sao cho tổng điểm là lớn nhất.

**Input Format**

Dòng 1 gồm số nguyên dương N và M; N dòng tiếp theo, mỗi dòng gồm M số nguyên A[i][j];

**Constraints**

1<=N,M<=500; 1<=A[i][j]<=1000;

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

5 5

1 0 31 5 25

28 26 32 46 7

26 40 36 13 16

7 26 14 6 11

42 45 11 10 21

**Sample Output 0**

208

**[DP]. Bài 19. Đếm số đường đi trong mê cung**

Cho mê cung được mô tả bởi một ma trận kí tự gồm N hàng và N cột. Mỗi ô trên ma trận chứa kí tự '.' tương ứng với đường đi và dấu \* tương ứng với bẫy. Một con chuột muốn đi từ ô (1, 1) tới ô (N, N) và chỉ được di chuyển khi một ô nào đó là đường đi và nó được di chuyển sang phải hoặc xuống dưới. Hãy đếm số cách con chuột có thể di chuyển tới đích. Vì kết quả quá lớn nên hãy lấy dư với 10^9 + 7

**Input Format**

Dòng đầu tiên là N; N dòng tiếp theo mỗi dòng là N kí tự

**Constraints**

1<=N<=1000

**Output Format**

Số đường đi tối đa

**Sample Input 0**

4

....

.\*..

...\*

\*...

**Sample Output 0**

3

**[DP]. Bài 20. Maximum Square SPOJ**

Bạn được cung cấp một ma trận nhị phân có N dòng và M cột chỉ bao gồm các số 0 và 1, hãy tìm hình vuông lớn nhất trong ma trận mà chỉ chứa toàn số 1.

**Input Format**

Dòng đầu tiên gồm 2 số N và M; N dòng tiếp theo mỗi dòng M số nguyên;

**Constraints**

1<=N,M<=500; 0<=A[i][j]<=1;

**Output Format**

In ra kết quả đáp án của bài toán

**Sample Input 0**

4 4

1 1 0 0

1 1 1 1

1 1 0 1

1 0 1 0

**Sample Output 0**

2

**[DP]. Bài 21. Distinct sum**

Cho mảng A[] gồm N phần tử là số nguyên dương, xét tất cả các tập con của A[] và tính tổng các phần tử trong tập này (bao gồm cả tập rỗng), sau đó liệt kê các tổng khác nhau có thể tạo thành theo thứ tự tăng dần.

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử của mảng A[]

**Constraints**

1<=N<=100; 1<=A[i]<=100;

**Output Format**

In ra các tổng có thể tạo thành từ các tập con của A[]

**Sample Input 0**

3

4 1 2

**Sample Output 0**

0 1 2 3 4 5 6 7

**[DP]. Bài 35. Odd-even**

Bạn được cung cấp một mảng A gồm N giá trị nguyên dương. Một mảng con của mảng này được gọi là mảng con Chẵn lẻ nếu số lượng các số nguyên lẻ trong mảng con này bằng số lượng các số nguyên chẵn trong mảng con này.

Tìm số mảng con Lẻ-Chẵn cho mảng đã cho.

**Input Format**

Dòng đầu chứa N; Dòng thứ 2 chứa N số nguyên trong dãy

**Constraints**

1<=N<=2.10^5; 1<=A[i]<=10^9;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

4

1 2 1 2

**Sample Output 0**

4

**[DP]. Bài 22. Sum substring**

Cho một số tự nhiên N được biểu diễn như một xâu kí tự, bạn hãy tính tổng của tất cả các số tạo bởi các xâu con liên tiếp của N, ví dụ N = 235 thì ta có tổng = 2 + 3 + 5 + 23 + 35 + 235.

**Input Format**

Dòng duy nhất chứa số nguyên dương N

**Constraints**

1<=N<=10^12

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

1807

**Sample Output 0**

2915

**[DP]. Bài 23. Xâu con đối xứng dài nhất**

Cho xâu S chỉ bao gồm các ký tự viết thường và dài không quá 1000 ký tự. Hãy tìm xâu con đối xứng dài nhất của S.

**Input Format**

Dòng duy nhất chứa xâu S

**Constraints**

1<=len(S)<=1000;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

edhcgeehahbbeggfcgcchffbffcgfghgc

**Sample Output 0**

5

**[DP]. Bài 24. Tổng bình phương**

Mọi số nguyên dương N đều có thể phân tích thành tổng các bình phương của các số nhỏ hơn N. Ví dụ số 100 = 10^2 hoặc 100 = 5^2 + 5^2 + 5^2 + 5^2. Cho số nguyên dương N. Nhiệm vụ của bạn là tìm số lượng ít nhất các số nhỏ hơn N mà có tổng bình phương bằng N.

**Input Format**

Một số tự nhiên N được viết trên 1 dòng.

**Constraints**

1≤N≤10000;

**Output Format**

Đưa ra kết quả của bài toán

**Sample Input 0**

100

**Sample Output 0**

1

**[DP]. Bài 25. Biến đổi xâu**

Cho hai xâu ký tự str1, str2 bao gồm các ký tự in thường và các thao tác dưới đây: Insert: chèn một ký tự bất kỳ vào str1. Delete: loại bỏ một ký tự bất kỳ trong str1. Replace: thay một ký tự bất kỳ trong str1. Nhiệm vụ của bạn là đếm số các phép Insert, Delete, Replace ít nhất thực hiện trên str1 để trở thành str2.

**Input Format**

Dòng 1 là xâu str1; Dòng 2 là xâu str2;

**Constraints**

1<=len(str1), len(str2)<=100;

**Output Format**

Đưa ra kết quả của bài toán

**Sample Input 0**

geek gesek

**Sample Output 0**

1

**[DP]. Bài 26. LIS 2**

Bạn được cung cấp một mảng chứa n số nguyên. Nhiệm vụ của bạn là xác định dãy con dài nhất tăng dần trong mảng, tức là dãy con dài nhất mà mọi phần tử đều lớn hơn phần tử trước đó. Một dãy con là một dãy có thể được dẫn xuất từ mảng bằng cách xóa một số phần tử mà không làm thay đổi thứ tự của các phần tử còn lại.

**Input Format**

Dòng đầu tiên chứa số nguyên n: kích thước của mảng. Sau đó có n số nguyên x1, x2,…, xn: nội dung của mảng.

**Constraints**

1≤n≤2⋅10^5; 1≤xi≤10^9;

**Output Format**

In ra chiều dài của dãy con tăng dài nhất

**Sample Input 0**

6

1 2 4 1 5 2

**Sample Output 0**

4

**Sample Input 1**

5

1 1 1 1 1

**Sample Output 1**

1

**[DP]. Bài 34. Cửa hàng hoa hồng**

Có hoa hồng trong một cửa hàng. Mỗi bông hồng được chỉ định một ID. Những bông hồng này được sắp xếp theo thứ tự 1, 2, 3... n. Mỗi bông hồng có một hệ số mùi được ký hiệu là smell[i] . Bạn muốn mua hoa hồng ở cửa hàng này với điều kiện phải mua hoa hồng theo phân khúc. Nói cách khác, bạn có thể mua hoa hồng từ đến l đến r. Bạn có thể loại bỏ nhiều nhất một hoa hồng khỏi phân đoạn hoa hồng này. Như vậy, chiều dài cuối cùng của hoa hồng là n hoặc n - 1. Nhiệm vụ của bạn là tính toán độ dài tối đa có thể có của dãy liền kề tăng dần các hệ số mùi của những bông hoa hồng này.

**Input Format**

Dòng 1 chứa số nguyên n; Dòng thứ 2 chứa n số smell[1], smell[2], .. smell[n]

**Constraints**

2<=n<=2.10^5; 1<=smell[i]<=10^9

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

5

1 2 5 3 4

**Sample Output 0**

4

**[DP]. Bài 27. Xóa chữ số**

Bạn được cung cấp một số nguyên n. Trên mỗi bước, bạn có thể trừ một trong các chữ số khỏi số. Cần thực hiện bao nhiêu bước để số đó bằng 0?

**Input Format**

Dòng duy nhất chứa số nguyên n

**Constraints**

1<=n<=10^6;

**Output Format**

In ra số bước tối thiểu

**Sample Input 0**

4

**Sample Output 0**

1

**Sample Input 1**

27

**Sample Output 1**

5

**Explanation 1**

Các bước thực hiện : 27→20→18→10→9→0

**[DP]. Bài 28. Select Array**

Bạn biết rằng một mảng có n số nguyên chỉ gồm các số từ 1 đến m và độ lệch giữa 2 phần tử liền kề trong mảng không được vượt quá 1 đơn vị. Bài toán đặt ra đó là cho bạn một mảng trong đó một số giá trị trong mảng chưa được xác định giá trị, nhiệm vụ của bạn là đếm số mảng phù hợp với mô tả, đó là các số liền kề trong mảng không lệch nhau quá 1 đơn vị và các giá trị trong mảng chỉ được nằm trong đoạn từ 1 tới m.

**Input Format**

Dòng nhập đầu tiên có hai số nguyên n và m: kích thước mảng và giới hạn trên cho mỗi giá trị.

Dòng tiếp theo có n số nguyên x1, x2,…, xn: nội dung của mảng. Giá trị 0 biểu thị một giá trị không xác định.

**Constraints**

1≤n≤10^5;  
1≤m≤100; 0≤xi≤m;

**Output Format**

In ra số lượng mảng phù hợp sau khi chia dư cho 1e9 + 7

**Sample Input 0**

3 5

2 0 2

**Sample Output 0**

3

**Explanation 0**

Các mảng có thể thỏa mãn : [2, 1, 2], [2, 2, 2], [2, 3, 2]

**[DP]. Bài 29. Equal set**

Nhiệm vụ của bạn là đếm số cách các số 1,2,…, n có thể chia thành hai tập có tổng bằng nhau. Các phần tử trong tập không xét đến thứ tự Ví dụ, nếu n = 7, có bốn nghiệm: {1,3,4,6} và {2,5,7}. {1,2,5,6} và {3,4,7}. {1,2,4,7} và {3,5,6}. {1,6,7} và {2,3,4,5}.

**Input Format**

Dòng duy nhất chứa số nguyên dương n

**Constraints**

1<=n<=500

**Output Format**

In ra kết quả sau khi chia dư với 10^9 + 7

**Sample Input 0**

7

**Sample Output 0**

4

**[DP]. Bài 30. Cắt hình chữ nhật**

Cho một hình chữ nhật a × b, nhiệm vụ của bạn là cắt nó thành các hình vuông. Trên mỗi lần cắt, bạn có thể chọn một hình chữ nhật và cắt nó thành hai hình chữ nhật sao cho tất cả độ dài các cạnh vẫn là số nguyên. Số lần di cắt tối thiểu có thể là bao nhiêu?

**Input Format**

Dòng duy nhất chứa 2 số nguyên a và b.

**Constraints**

1<=a,b<=500;

**Output Format**

In ra số lần cắt tối thiểu

**Sample Input 0**

3 5

**Sample Output 0**

3

**[DP]. Bài 31. Scheduling**

Có n dự án bạn có thể tham dự. Đối với mỗi dự án, bạn biết ngày bắt đầu và ngày kết thúc của dự án và số tiền bạn sẽ nhận được khi hoàn thành nó. Bạn chỉ có thể tham dự một dự án trong một ngày. Số tiền tối đa bạn có thể kiếm được là bao nhiêu?

**Input Format**

Dòng đầu tiên chứa số nguyên n: số lượng dự án. Sau đó, có n dòng. Mỗi dòng như vậy có ba số nguyên ai, bi và pi: ngày bắt đầu, ngày kết thúc và tiền thưởng.

**Constraints**

1≤n≤2⋅10^5; 1≤ai≤bi≤10^9; 1≤pi≤10^9;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

4

2 4 4

3 6 6

6 8 2

5 7 3

**Sample Output 0**

7

**[DP]. Bài 33. Lát gạch**

Cho một bảng hình chữ nhật cỡ nxm hình vuông đơn vị, nhiệm vụ của bạn là đếm số cách lấp đầy hình chữ nhật này bằng cách viên gạch có cỡ 1x2 và 2x1.

**Input Format**

Dòng duy nhất chứa 2 số n và m.

**Constraints**

1≤n≤10; 1≤m≤1000;

**Output Format**

In ra đáp án của bài toán chia dư với 1e9 + 7

**Sample Input 0**

4 7

**Sample Output 0**

781

**[Mảng 2 Chiều]. Bài 19. Maximum path sum**

Cho ma trận A các số nguyên có N hàng và M cột. Tìm đường đi từ ở [1, 1] tới ô [N, M] sao cho tổng các số trên đường đi là lớn nhất có thể, biết rằng ở mỗi bước chỉ có thể đi từ ô hiện tại xuống ô phía dưới hoặc đi sang phải.

**Input Format**

Dòng đầu tiên N và M. N dòng tiếp theo mỗi dòng gồm M phần tử.

**Constraints**

1≤N,M≤100; 1≤A[i][j]≤10^9

**Output Format**

In ra đường đi có tổng lớn nhất.

**Sample Input 0**

3 3

1 2 2

3 10 2

5 7 2

**Sample Output 0**

23

**Explanation 0**

Giải thích : Đường đi được chọn (1, 1) -> (2, 1) -> (2, 2) -> (3, 2) -> (3, 3)

**[Mảng 2 Chiều]. Bài 26. Maximum path sum 2**

Cho mảng 2 chiều A gồm N hàng và N cột, hãy tìm 1 đường đi từ dòng 1 tới dòng N sao cho các phần tử trên đường đi đó là lớn nhất có thể. Biết rằng ở mỗi bước đi từ ô hiện tại chỉ có thể đi xuống ô dưới bên trái, ô dưới bên phải hoặc ô dưới của ô hiện tại. Hãy tìm 1 đường đi có tổng các số trên đường đi là lớn nhất.

**Input Format**

* Dòng 1 là N
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 1<=N<=100
* 1<=A[i][j]<=100

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

5

5 1 8 6 7

6 0 0 3 7

3 4 1 5 3

9 8 9 2 2

5 3 4 5 3

**Sample Output 0**

33

**Explanation 0**

Đường đi có tổng lớn nhất : 7 + 7 + 5 + 9 + 5 = 33

**[Hàm, Lý Thuyết Số]. Bài 36. Tam giác pascal**

In ra tam giác pascal với chiều cao là n. Nếu các giá trị của tam giác pascal quá lớn, in ra giá trị trong tam giác này lấy dư với 1e9+7

**Input Format**

Số nguyên dương n

**Constraints**

1≤n≤10^2

**Output Format**

Tam giác pascal

**Sample Input 0**

6

**Sample Output 0**

1

1 1

1 2 1

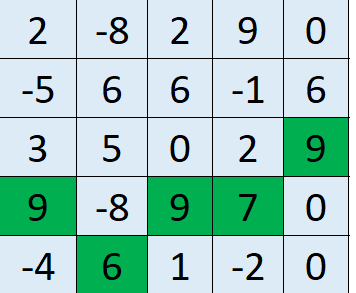
1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

**[Mảng 2 Chiều]. Bài 27. Maximum path sum 3**

Cho mảng 2 chiều A gồm N hàng và N cột, hàng được đánh số từ 1 đến N từ trên xuống dưới, cột được đánh số từ 1 tới N từ trái sang phải, hãy tìm 1 đường đi từ một ô ở cột 1 tới 1 ổ ở cột N sao cho các phần tử trên đường đi đó là lớn nhất có thể. Biết rằng ở mỗi bước đi từ ô hiện tại (i, j) chỉ có thể đi sang phải (i, j + 1) hoặc đi xuống ô dưới bên phải (i + 1, j + 1), ô bên trên bên phải(i - 1, j + 1). Hãy tìm 1 đường đi có tổng các số trên đường đi là lớn nhất.



**Input Format**

* Dòng 1 là N
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 1<=N<=100
* -100<=A[i][j]<=100

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

5

2 -8 2 9 0

-5 6 6 -1 6

3 5 0 2 9

9 -8 9 7 0

-4 6 1 -2 0

**Sample Output 0**

40

**Explanation 0**

Đường đi có tổng lớn nhất : 9 + 6 + 9 + 7 + 9 = 40