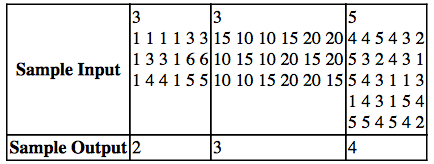
1/Trò chơi chẵn lẻ là trò chơi hai đối thủ được mô tả như sau: Xuất phát từ bảng trò chơi là một bảng vuông kích thước n × n gồm n dòng và n cột. Các dòng của bảng được đánh số từ 1 đến n, từ trên xuống dưới. Các cột của bảng được đánh số từ 1 đến n, từ trái qua phải. Trên mỗi ô của bảng ghi một số nguyên. Hai đối thủ luân phiên thực hiện nước đi. Đối thủ đến lượt chơi của mình được phép xoá dòng cuối cùng nếu tổng các số trên dòng đó là số chẵn hoặc là cột cuối cùng nếu tổng các số trên cột đó là số chẵn.  
Đối thủ thắng cuộc là người xoá được ô cuối cùng của bảng hoặc sau khi thực hiện nước đi của mình thì tổng các số trên dòng cuối cùng và tổng các số trên cột cuối cùng của bảng đều là số lẻ.  
Yêu cầu: Cho biết bảng số của trò chơi, hãy xác định xem người đi trước có cách chơi giành phần thắng hay không?  
Dữ liệu: Vào từ file văn bản PARIGAME.INP:  
• Dòng thứ nhất chứa số nguyên dương k là số lượng bộ dữ liệu;  
• Tiếp theo là k nhóm dòng, mỗi nhóm dòng tương ứng với một bộ dữ liệu có dạng:  
o Dòng thứ nhất chứa số nguyên dương n (n ≤ 500).  
o Dòng thứ i trong số n dòng tiếp theo chứa n số nguyên dương (mỗi số không vượt  
quá 109) là các số trên dòng thứ i của bảng trò chơi, i = 1, 2, ..., n.  
Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.  
Kết quả: Ghi ra file văn bản PARIGAME.OUT gồm k dòng, mỗi dòng là kết quả tương ứng với  
một bộ dữ liệu theo thứ tự xuất hiện trong file dữ liệu vào: ghi thông báo ‘YES’ nếu người đi trước

2/Cho số nguyên dương N. Hãy cho biết có thể phân tích N thành tổng các số tự nhiên liên tiếp hay không? Nếu có hãy phân tích N thành tổng của các số tự nhiên liên tiếp với nhiều số hạng nhất có thể được.  
Dữ liệu vào: cho trong File PTICH.INP gồm một số dòng, mỗi dòng ghi một số nguyên dương N (N <= 10^9)  
Kết quả: ghi ra file PTICH.OUT theo cấu trúc sau:  
+ Nếu N phân tích được thành tổng các số tự nhiên liên tiếp thì ghi chữ d sau đó là các số hạng trong cách phân tích đó, các số hạng ghi cách nhau một dấu cách, mỗi dòng ghi không quá 25 số.  
+ Nếu N không phân tích được thì chỉ ghi chữ k.  
Ví dụ:  
  
PTICH.INP  
12  
8  
PTICH.OUT  
d: 3 4 5  
k

A long time ago in a three-dimensional space far away, a tribe of rectangles lived happily. The rectangles lived a spiritual life, parallel with one of the coordinate planes.  
  
One day, a cuboid walked into their small world, riding steadily on an icosahedron, showing off its sharp corners and positive volume. The rectangles watched in awe and dreamed of being cuboids.  
  
Nothing would ever be the same from that day on. The rectangles started comparing each other by area, perimeter and even by the ratio of the lengths of their sides.  
  
Soon the first conflict ensued over the ownership of shared points. In time, each pair of rectangles sharing at least one point (including those merely touching each other) got into a conflict and became enemies.  
  
It is up to you to restore peace in the community, by meeting with every pair of rectangles in conflict.  
  
Write a program that finds how many such pairs there are.  
  
INPUT  
  
The first line of input contains the integer N (1 ≤ N ≤ 100 000), the number of rectangles.  
  
Each of the following N lines contains 6 integers separated by single spaces. The first three numbers represent the coordinates of one corner of the rectangle, the other three are the coordinates of the opposite corner.  
  
The coordinates are integers between 1 and 999 (inclusive).  
  
Each rectangle is parallel to one of the coordinate planes, meaning that in exactly one of the three dimensions, the two corresponding coordinates will be equal.  
  
OUTPUT  
  
Output the total number of rectangles in conflict on a single line.  
  
  


Joe is given a sheet of paper marked with n small pieces to cut. As he is a person of optimization and efficiency, he seeks to cut the paper into n separate pieces in the least physical (or the most energy-saving) way possible. Let's help Joe.  
  
Given that:  
Piece number i, having starting point l and ending point r, is cut at m (cutting point).  
For one single cut from (l -> r) into (l -> m) and (m+1 -> r), it takes an amount of energy equivalent to a[m]\*(r-l+1).  
a[m] is the thickness of piece number m.  
  
Constraints:  
+ n <= 300.  
+ Thickness of paper <= 100.  
  
Input:  
+ The first line contains an integer n, followed by a line of n-1 numbers (\*) representing the thickness of first n-1 pieces. ((\*) Note: it only takes one cut for the last two pieces).  
  
Output:  
+ Display the lowest possible energy spent.  
  
Input  
5  
2 3 1 4  
  
Output  
25  
  
Explanation:  
+ Step 1: To cut the paper from (1 -> 5) into (1 -> 3) & (4 -> 5), energy of 1\*5 is spent  
+ Step 2: To cut the piece from (1 -> 3) into (1) & (2 -> 3), energy of 2\*3 is spent  
+ Step 3: To cut the piece from (2 -> 3) into (2) & (3), energy of 3\*2 is spent  
+ Step 4: To cut the piece from (4 -> 5) into (4) and (5), energy of 4\*2 is spent  
=> Total energy spent: 25