Problem 1. Sum of Odd Numbers 數字區間奇數和

(Time Limit: 2 seconds)

問題描述 :

Given a range *a* to *b*, please find the sum of all the odd numbers within the range. For example, the sum of the odd numbers within the range [1,6] is 1 + 3 + 5 = 9.

設定數字區間 a 到 b，請在 a，b 區間內找到所有奇數之和。 例如，區間的範圍為[1,6]，則此區間的奇數和是 1 + 3 + 5 = 9。

輸入說明:

Given a number of datasets, where each dataset contains two numbers indicating the range, namely a and b (0 ≦ a ≦ b ≦ 100).

輸入若干組資料，每組資料均包含表達區間範圍的兩個數字，a 和 b（0 ≦ a ≦ b ≦ 100）。

輸出說明：

Output the sum of odd numbers within the range of each dataset.

請輸出每組資料其區間內所有奇數之和。

範例:

Sample Input:	**Sample Output:**
1 5	9
2 4	3
3 7	15
10 11	11
1 1	1

Problem 2. Ten Half 十點半

(Time Limit: 2 seconds)

問題描述 :

"Ten Half" is a simple poker game similar to BlackJack. A is one point, 2 is two points, …, 10 is ten points, and each of J, Q, K is a half point. If your total point of cards is over 10.5, you lose the game.

"十點半"是一款類似於二十一點的簡單撲克遊戲。A 是 1 分,2 是 2 分,...,10 是 10 分,J,Q,K 分別是 0.5 分。 如果您的總積分超過 10.5,您就輸了這場遊戲。

輸入說明:

The first line is an integer n (0<n<=10) that indicates the number of test cases. Other lines contain the cards you have in each test case. There is at least one card in each test case.

第一行是整數 n(0 <n <= 10),表示測試例子的數量。 其他行的每一行分別包含一個測試例子,代表您已拿到的撲克牌。 每個測試例子中至少有一張撲克牌。

輸出說明:

Output the probability of losing the game (A%), where A is a rounded integer scaled from 0 to 100 after you draw the next card. If the initial cards in the test case have already exceeded 10.5 points, output -1. Add a newline character at the end of each output.

輸出你再抽下一張牌後,輸掉遊戲的百分比機率(A%),其中 A 為四捨五入至 0 到 100 的整數。 如果測試例子中原本的牌已經超過 10.5 點,則輸出 -1。在每個輸出的末端請加上換行字元。

範例:

Sample Input:	**Sample Output:**
2 A 3 5 4 6 J K	69 -1

Problem 3. The Determinant 行列式求值

(Time Limit: 2 seconds)

問題描述 :

A determinant is defined as:

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

= a * e * i + d * h * c + g * b * f - c * e * g - b * d * i - a * f * h

Calculate a determinant of 3x3 matrices.

已知行列式運算為

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

= a * e * i + d * h * c + g * b * f - c * e * g - b * d * i - a * f * h

求上述三階行列式的結果。

輸入說明:

The first line of the input contains an integer N indicating the number of test cases ($1 \le N \le 100$). For each test case, there are three lines with three integers separated by spaces.

For example, a test case is following:

1 2 3

4 5 6

7 8 9

第一行輸入一正整數 N，表示共有 N 筆測試資料($1 \le N \le 100$)，每筆測試資料為一個行列式，行列式內容為 3 列，每一列有 3 個整數，用空白分開。

例如輸入測試內容為：

1 2 3

4 5 6
7 8 9

輸出說明:

Output the answer of each test case. Add a “newline” in the end of the output.

輸出計算的答案，最後必須有換行字元。

範例:

Sample Input:	**Sample Output:**
2 1 2 3 4 5 6 7 8 9 6 9 4 2 4 9 1 8 3	0 -285

Problem 4. Multiplication 數數之積

(Time Limit: 2 seconds)

問題描述 :

Given a dataset of integers, calculate the product of these numbers by multiplication.

給定若干整數，輸出其乘積。

輸入說明:

Input at most nine lines of test datasets, and each dataset contains n integers ($1 \le n \le 12$) with the range m ($0 \le m \le 40$).

The integers are separated by spaces.

輸入最多 9 筆測資，每一列有一筆測試資料。每筆測試資料包含 n 個整數($1 \le n \le 12$)；其整數範圍為 m ($0 \le m \le 40$)，整數之間以空白隔開。

輸出說明:

For each input line, print the product value of those numbers involved. Please print a cartridge return at the end of each output line.

請針對每筆測資輸出其所有整數之乘積，最後必須有換行字元。

範例:

Sample Input:	**Sample Output:**
2 3 4 1 2 2 4 0 7	24 0

Problem 5. 18 啦遊戲設計

Sip-Pat-La

(Time Limit: 2 seconds)

問題描述 :

" Sip-Pat-La" is one of the popular dice games in Taiwan.
Players roll four dice in the game and score the results. There are three conditions:

1. Win-take-all: Four dice have the same number;
2. Meaningless: Thee dice have the same number, or four dice have different numbers;
3. Score: Two dice have the same number; the score is the sum of the rest two dice. If the four numbers are two pairs, select the pair with the bigger number to calculate the score.

Please write a program and judge the result according to the input numbers.

有一種坊間常見的擲骰子遊戲稱為 "18 啦 "~
其玩法是玩家擲四個骰子來算分，會產生有三種結果：

1. 通殺：四 顆骰子點數均相同；
2. 無意義：三顆骰子點數相同，或四顆骰子點數均不相同；
3. 計分：兩個骰子相同點數，分數即為另兩個骰子點數和；若四個骰子的點數兩兩相同形成兩對，分數選擇較多點數和的那一對骰子。

請寫一程式，由輸入四顆骰子點數來判斷結果。

輸入說明:

Enter the four numbers of dice points per line, where the numbers range from 1 to 6.
每行輸入四個骰出的點數，輸入的點數介於 1~6 之間 。

輸出說明:

The results of each measurement are output in order. If the condition is ~~a~~ winner-take-all, the uppercase English letter WIN is displayed. If it is meaningless, the uppercase English letter R is displayed. If the points can be calculated, show the score. No spaces are required before and after the output, and there must be a newline character at the end.

依序輸出各筆測資的結果，若是通殺，則顯示大寫英文字母 WIN ， 若是無意義 ，則顯示大寫英文字母 R ，若可計算點數，則顯示最後點數 。 輸出結果前後均不需留空格 ，最後必須有換行字元。

範例:

Sample Input:	**Sample Output:**
2 3 4 5	R
3 4 3 4	8

Problem 6. Covering a Hole

覆蓋孔洞

(Time Limit: 3 seconds)

問題描述：

Tom works in a company that produces covers for all kinds of holes, such as holes on streets and wells. He encounters a problem as follows: given a hole H which is a polygon with interior angles of only 90 or 270 degrees, determine the smallest rectangular cover that can completely cover H. In this problem, H is given in a coordinate system such that each of its edges is either vertical or horizontal. When covering a hole, each edge of the cover should also be either vertical or horizontal in the same coordinate system.

Consider the example in Figure 1. It is easy to see that the smallest rectangular cover that can completely cover H is a rectangle of size 4 × 8.

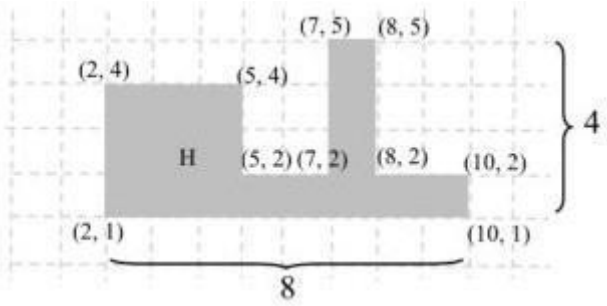

Figure 1: a rectangular hole H.

In this problem, you are asked to find the area of the smallest rectangular cover that can completely cover H. For example, in Figure 1, the output is 32.

Technical Specification

1. The number of the vertices of H, denoted by n, is a positive integer between 4 and 100.
2. The x-coordinates and y-coordinates of vertices are integers between 0 and 1000.

湯姆在一家公司工作，該公司生產各種孔洞的蓋子，例如街道和水井的孔洞。他遇到以下問題：給定一個孔洞 H（該孔洞是一個僅具有 90 度或 270 度內角的多邊形），請找出可以完全蓋住 H 的最小矩形蓋子。在此問題中，H 的坐標是

由每個邊緣均為垂直或水平的坐標系統表示。覆蓋孔洞時，蓋子的每個邊緣在同一坐標系統中也應為垂直或水平。
考慮圖 1 中的示例。很容易看出，可以完全覆蓋 H 的最小矩形蓋子是大小為 4 × 8 的矩形。

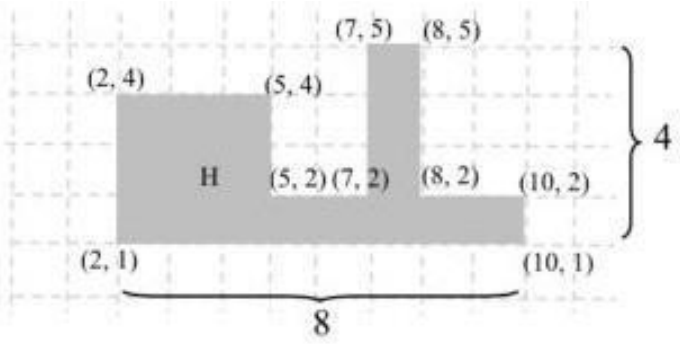

圖 1：矩型孔洞 H

在此問題中，你被要求找出可以完全覆蓋 H 的最小矩形蓋子。例如，在圖 1 中，輸出為 32。

特別定義：
1. H 的頂點數（用 n 表示）是 4 到 100 之間的正整數。
2. 頂點的 x 坐標和 y 坐標是 0 到 1000 之間的整數。

輸入說明:

The first line is an integer t, $1 \le t \le 10$, indicating the number of test cases.
Each test case starts with one line containing the number n, $4 \le n \le 100$, of vertices of the hole H. Then, n lines follow, each of which includes two integers x and y, $0 \le x, y \le 1000$, which are the coordinates of the vertices of the hole's polygon. In the order, they would be visited on a trip around the polygon.

第一行是整數 t，其中 $1 \le t \le 10$，表示測試案例的數量。每個測試案例開始的第一行為整數 n（$4 \le n \le 100$），代表孔洞 H 的頂點數量 。接著是 n 行數值，每行包含兩個整數 x 和 y，其中 $0 \le x, y \le 1000$，表示孔洞多邊形的頂點坐標，依照在多邊形周圍繞一圈的順序讀取它們。

輸出說明:

For each test case, output the area of the smallest rectangular cover that can completely cover H in one line.

對於每個測試案例，輸出一行可以完全覆蓋 H 的最小矩形蓋子的面積。

範例:

Sample Input:	**Sample Output:**
2 10 10 1 10 2 8 2 8 5 7 5 7 2 5 2 5 4 2 4 2 1 4 2 1 5 1 5 4 2 4	32 9