

Problem 1. 18 啦遊戲設計

(Time Limit: 2 seconds)

問題描述：

有一種仿間常見的遊戲稱為 "18 啦"。玩法介紹：由四個骰子來擲，只要其中任兩個是相同點，才開始計算點數，點數即為另兩個點數相加，若另兩個也一樣，則取較大的一組相加，另外有下列 2 種特殊情況：

1. 擲出 4 顆骰子點數均相同，稱為通殺。
2. 任三顆點數相同或四顆點數均不相同，即為無意義。

請寫一程式，個別輸入四顆骰子點數，判斷結果。

輸入說明：

有多筆測資,每筆測資輸入四次骰出的點數，一行輸入一次點數，輸入的點數為 1~6 之間。

輸出說明：

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。

範例：

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

Problem 2. 線段切割

(Time Limit: 1 second)

問題描述：

給定一條直線線段，要將該線段切割成若干不同長度的線段。請問最多可以切割成多少不同長度的線段？切割前後的線段長度均以公分為基本單位。例如 6 公分的線段最多可以切割成 3 段： $6 = 1 + 2 + 3$ 。

輸入說明：

輸入的測試資料包含若干行，每一行包含一個正整數 n ， $1 \leq n \leq 2^{32} - 1$ ，代表原始線段長度。

輸出說明：

對於每一筆測試資料，列印最大切割個數在一行中。

範例：

Sample Input:	Sample Output:
6	3
45	9
60	10

Problem 3. 容器量水

(Time Limit: 2 seconds)

問題描述：

在終極警探 (Die Hard) 電影中，歹徒在噴水池前放置一個定時炸彈以及兩個沒有刻度的容器，容器裝滿分別可裝五公升和三公升的水。歹徒留下線索，只要將四公升的水放置於定時炸彈上，則炸彈就會停止計時並解除。兩位警探一陣手忙腳亂，很幸運的量出了四公升的水，解除了危機。有個問題，我們怎麼知道現場的容器一定可以量出目標水量？假設現場有好幾個容量不同的容器，請寫程式判斷是否能利用其中的某些容器準確的量出歹徒所設定的目標水量？

輸入說明：

程式輸入的包含若干筆測資，每筆測資的第一列包含兩個整數 n 和 k ，數字間以一個空格隔開，其中 n 代表容器的個數， $1 \leq n \leq 100$ ； k 則是代表目標水量。第二列有 n 個整數，整數大小不超過 100，數字間以一個空格隔開，代表容器的容量。

輸出說明：

如果可以量出目標水量，則輸出 “YES”，否則輸出 “NO”。

範例：

Sample Input:	Sample Output:
2 4	YES
3 5	NO
3 5	
2 4 6	

Problem 4. 狀態機 (State Machine)

(Time Limit: 3 seconds)

問題描述：

給定一狀態機 (State Machine) 與該狀態機的初始狀態 (Initial State)，接著讀入一輸入字串 (Input String)，則狀態機便會依照字串中的每個字元指示，進行狀態轉換；請輸出此狀態機之狀態運轉過程。

舉例來說，圖 1 表示一狀態機，各個狀態之間，會依連線輸入字元與連線方向之指示進行狀態轉換。假定初始狀態為 2 號，輸入之字串為 "c#b\$#bac"，則該狀態機之狀態運轉過程為 "232132123" (答案包含初始狀態)。

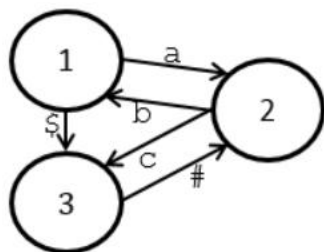


圖 1

圖 1 之狀態機可用 3×3 矩陣形式表示：
$$\begin{bmatrix} - & a & \$ \\ b & - & c \\ - & \# & - \end{bmatrix}$$
，其中，矩陣之第 r

列第 c 行元素，表示由狀態 r 轉換至狀態 c 之輸入字元，若其字元符號為 "-"，則表示狀態 r 至狀態 c 之間沒有連線。

輸入說明：

第一列包含兩個整數，以空格隔開，分別表示狀態機之狀態數目 n 與初始狀態編號 s ，其中 $s \leq n \leq 20$ 。第二列為字串輸入，最多包含 100 個字元。接下來 n 列為 $n \times n$ 之陣列輸入，用來表示該狀態機之結構；每列有 n 個元素，以空格隔開。

輸出說明：

輸出該狀態機之狀態運轉過程，答案需包含初始狀態，最後以換行作結束。

範例：

Sample Input:	Sample Output:
---------------	----------------

3 2 c#b\$#bac - a \$ b - c - # -	232132123
--	-----------

Problem 5. Number Maze

(Time Limit: 5 seconds)

Problem Description

We are given an m by n matrix (m rows and n columns) that denotes a maze, where $1 < m, n < 10$. Each grid of the matrix contains a positive integer number. Suppose that we need to walk from the left-top corner (matrix index is $[0][0]$) to the right-bottom corner (matrix index is $[m-1][n-1]$). Each walk can follow the directions of up, down, left, and right without crossing the boundary. Please write a program to find the walk path with the constraint that the summation of numbers contained in these visited grids is minimum. Your program needs to output the minimum summation. For the following example, the output result is 30.

1 →	2 →	3 →	4 ↓
5	6	7	8 ↓
9	10	11	12 ↓

Input Format

The first line contains two integer numbers separated by a whitespace and these two numbers denote the number of rows and the number of columns of the matrix, respectively. Beginning from the second line, each line indicates the content (a series of numbers separated by a whitespace) of each row.

Output Format

Output the minimum summation of numbers visited along the path from the entry point to the exit point.

Example

Sample Input:	Sample Output:
4 4 1 2 3 0 3 1 5 7 2 6 8 1 5 7 1 5 3 5 1 2 3 4 5 5 4 3 2 1	19 11

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