

Problem 1. 邊境防禦

(Time Limit: 1 seconds)

問題描述:

汪徹使用空拍機飛到上收集週邊地形數據，將拍到的圖片轉換為一二元圖，其中用 1 表示土地，以 0 表示海洋湖泊。之後在二元圖中找出其邊境，請你寫一個程式讀入二元圖數據，之後輸出邊境數據圖。

輸入說明

第一行為一個正整數 N ($1 \leq N \leq 5$) 代表共有幾組測試資料。之後接下來有 N 組數據，每組第一行為兩個正整數 n 及 m ，表示二元圖的大小為 $n \times m$ ，之後 n 行每行有 m 個數字 ($1 \leq n, m \leq 100$ ，每個數字後面都有一個空格字元)，分別是 0 或 1，即為該地形數據。

輸出說明:

輸出邊境數據圖，如果將為邊境的點以 0 表示，否則輸出底線"_"，每個點後面都有空格字元。每組測試資料結果與前一組之間空一行。

範例

Sample1 Input:	Sample1 Output:
2	-----
5 7	--_000_--
0000000	_0___0_
0011100	--_000_--
0111110	-----
0011100	
0000000	-----
10 10	_00000000_
0000000000	_0_____0_
0111111110	_0_____0_
0111111110	_0___0__0_
0111111110	_0__0_0_0_
0111111110	_0___0__0_
0111101110	_0_____0_

0111111110	_00000000_
0111111110	-----
0111111110	
0000000000	

Problem 2. 購票機

(Time Limit: 1 seconds)

問題描述:

K 歌之王演唱會門票開賣，假設門票的自動售票機只能接受 10 元、5 元、以及 1 元的硬幣，請問小美想買的演唱會位置之門票為 N 元時，所需的各種幣值硬幣最少的數量為多少？

輸入說明

輸入為一個正整數 N ($1 \leq N \leq 1,000,000$)，表示門票為 N 元。

輸出說明:

依序輸出 10 元、5 元、以及 1 元的硬幣最少的數量，中間用空格隔開。

範例

Sample1 Input:	Sample1 Output:
37	3 1 2

Problem 3. 訊息加密

(Time Limit: 1 seconds)

問題描述:

假設有一加密法，會將要傳送的訊息進行換位方式的加密。

假設現有一個字串"Hi,how are you!!"，我們可以將每個字元拆開分入 4*4 的矩陣內，如下表：

H	i	,	h
o	w		a
r	e		y
o	u	!	!

加密方式使用「行」的方式讀取將可得到"Horoiweu, !hay!"_{△△}

接收端收到加密字串後，再以行的方式回填至矩陣內，再使用「列」的方式讀取將可得到原文。

輸入說明

輸入共有 3 行。

第 1 行是 1 個整數，若為 0，代表傳送端，需要進行加密，若為 1，代表接收端，需要解密。

第 2 行是 1 個整數 N，表示矩陣的維度。如問題敘述，我們使用 4*4 的矩陣，所以維度值為 4， $(1 \leq N \leq 50)$ 。

第 3 行為一個字串，代表要加密或是解密的字串，字串長度 $\leq 2,500$ 。

輸出說明:

依照要求，輸出加密或解密後的字串。如字串前後有空白，請去除。

範例

Sample1 Input:	Sample1 Output:
0 4 Hi,how are you!!	Horoiweu, !hay! _{△△}

Problem 4. 商品排序

(Time Limit: 1 seconds)

問題描述:

嘉明收集完目前商品的特性資料，假設目前貨架上的商品有 5 個，嘉明想要針對商品的特性資料來進行排序，排序的方式是：

1. 先比較利潤，利潤愈高的商品排在最上面。
2. 若利潤相同，則比較成本，成本較高者，排在上面。
3. 若成本亦相同，則比較重量，若重量仍相同，最後比較年限。
4. 若兩者所有商品的全部數值均相同，則其依商品編號排序，小的排最上面。

請設計一個程式，按照上述規則進行排序。

輸入說明

輸入共 5 行，每 1 行表示 1 個商品的特性資料。

每 1 行共有 5 個正整數 A、B、C、D、E，A 表示商品編號(00001~99999)；B 表示商品利潤(0~300)；C 表示商品成本(0~100)；D 表示商品重量(0~100)；E 表示商品年限(0~100)；每個正整數間用逗號隔開。

輸出說明:

輸出根據排序規則進行排序的結果。

注意: 商品編號需要補 0 到 5 位數，例如 1 號需輸出為 00001。

範例

Sample1 Input:	Sample1 Output:
99001,155,33,55,67	99003,170,60,70,40
99002,170,48,59,63	99002,170,48,59,63
99003,170,60,70,40	99005,162,60,34,68
99004,157,55,34,68	99004,157,55,34,68
99005,162,60,34,68	99001,155,33,55,67

Problem 5. 制高點尋找股標相對高點

(Time Limit: 1 seconds)

問題描述:

后裔正在進行會戰前的戰地勘查，他想要找到該戰地相對的制高點，想事先佔據於制高點，想達到居高臨下的戰爭優勢，假設現在后裔從主帥那裡取得地形高度數據為：

[100,127,122,143,144]

相對制高點的定義是指，它的高度數據比前後都來得高。所以 127 及 144 都是相對制高點。后裔在找到各個相對制高點後，會將資訊回傳主帥，為了避免中途遭敵軍攔截，他會依索引的方式回覆，假設索引值由 0 開始，因此後裔需要回傳 1 以及 4 是相對高點。

輸入說明

第一行為一個正整數 N ($1 \leq N \leq 5$)，代表共有幾組測試資料。

之後接下來有 N 組數據，每組測試資料為 1 行，每行有 m 個正整數 ($5 \leq m \leq 200$)，表示取得地形高度的數據，每個整數用空白隔開。

每個地形高度的數據範圍為 1~5,000。

輸出說明:

輸出相對高點的索引值，由小到大排序。

範例

Sample1 Input:	Sample1 Output:
2 100 127 122 143 144 920 2443 164 46 2604 3274 44 1017 191 3957 2915 4959 2399 4546 3981 4117 2938 2049 4798 293 4416 2312 522	1 4 1 5 7 9 11 13 15 18 20

Problem 6. 產線排程

(Time Limit: 1 seconds)

問題描述:

工廠接單生產產品，每單都有其完成期限與優先性。例如，生產單 A 可能於時間 T 進來，在 T_d 之前必須完成此生產單，其生產所需要花費的時間為 T_e 。

今定義排程策略為：

1. 優先性數值越低優先次序較高。
2. 優先次序較高者有優先執行的權利；
3. 若優先次序相同，則比較兩者的完成期限，完成期限較早者有優先執行權。
4. 生產單必須完成正在生產的訂單後，才會再依相同策略，選擇下一個要生產的訂單。

輸入說明

第一行為一個正整數 n ($1 \leq n \leq 10$)，代表共有 n 個生產單。

其後有 n 行，每一行代表一個生產單。每一行之資料依序為單號 D (不超過 10 個字元)、到達時間 T ($0 \leq T \leq 10$)、生產時間 T_e ($1 \leq T_e \leq 10$)、完成期限 T_d ($1 \leq T_d \leq 100$)、及其優先性 p ($1 \leq p \leq 10$)。各項資料之間，以一個空白分隔。

注意：測資保證每一個工作均可在其完成期限之前完成，故毋需考慮無法完成工作之情形。請生產單並不一定按照到達時間的順序輸入。

輸出說明:

輸出依排程策略所排列出的工作執行順序。各項資料之間以一個空白分隔。

範例

Sample1 Input:	Sample1 Output:
4 A1 3 3 30 8 A2 6 4 25 4 A3 4 5 30 4 A4 8 3 30 1	A1 A2 A4 A3

Problem 7. 挖金礦

(Time Limit: 1 seconds)

問題描述:

阿夏在金礦山上挖礦，他將礦地畫成一塊 $N \times N$ 的正方形土地，每個礦點都有不同的礦產量，假設給定兩個座標來代表一矩陣範圍，請問在這矩陣範圍內的礦產量有多少呢？

輸入說明

第一行會有 2 個正整數 N, M ($N \leq 100, M \leq 100,000$)， N 表示正方形的礦地；

接下來一行有 $N \times N$ 個不超過 100 的正整數，依序代表每個礦點的礦產量。

接下來 M 行，每行有四個數字 $startX, startY, endX, endY$ ($1 \leq startX \leq endX \leq N, 1 \leq startY \leq endY \leq N$)，代表阿夏想要挖的矩陣範圍。

輸出說明:

請輸出每個矩陣範圍的礦產量，輸出一行後都需有換行。

範例

Sample1 Input:	Sample1 Output:
3 3	45
1 2 3	6
4 5 6	12
7 8 9	
1 1 3 3	
1 1 1 3	
1 1 3 1	

