

Problem 1. 文字及字母出現次數

(Time Limit: 2 seconds)

問題描述：

在電腦科學上，計算一串文字上各個字母出現的頻率是常被用到的技術，這對壓縮來講是很重要的資訊，而計算字數也可以幫助人們作校正的工具。一行文字被空白、逗點或是句點所分隔而形成很多字，例如 "I have a pencil." 這行字就有 I，have，a，pencil 這四個字，即此行字數為 4。所以現在要請你幫忙設計一個程式來計算一行文字的字數及各個字母出現的次數。

輸入說明：

輸入包含若干筆測資，每筆測資包含一行正常的英文文字，也就是不要有開頭是空白或是有連續兩個空白的情形發生，並且內容只能包含英文字母、空白、逗點、句點。注意：輸入的字串長度最多是 100。

輸出說明：

每筆測資的輸出：第一行輸出一個正整數 n，表示此行文字的字數。第二行開始依字典序輸出在此行文字中有出現的字母及出現的次數。注意：大小寫要分開計算，當大小寫同時存在，則先輸出大寫再輸出小寫。

字母個數輸出的格式為:字母、":","、個數，中間以一個空格隔開。

範例：

Sample Input:	Sample Output:
I had a dog How are you	4 a : 2 d : 2 g : 1 h : 1 I : 1 o : 1 3 a : 1 e : 1 H : 1 o : 2 r : 1

	u : 1 w : 1 y : 1
--	-------------------------

Problem 2. 貨船載貨最大數

(Time Limit: 1 second)

問題描述：

一艘貨船最大載重量為 W ，最多可以裝載 Y 個貨櫃，現在有 N 個貨櫃需要裝到貨船上，貨櫃編號為 $X_0, X_1 \sim X_{n-1}$ ，每個貨櫃的重量單位分別為 $k_0, k_1 \sim k_{n-1}$ ，在不超過最大載重量 W 下，及裝載剛好 Y 個貨櫃，如何裝載才能裝載最大的貨櫃重量到貨船上，請求其解。

輸入說明：

輸入包含若干筆測資，每筆測資包含一行資料，資料以逗號隔開：

第一筆資料為貨船最大載重量 ($0 < W < 500$)

第二筆資料為最多裝載幾個貨櫃 ($0 < Y < 100$)

第三筆資料為現在有幾個貨櫃 ($0 < N < 100$)

第四筆開始至最後共 N 個資料為 $k_0, k_1 \sim k_{n-1}$ 貨櫃重量。

輸出說明：

每筆測資輸出格式如下：輸出以 k 為代號，以逗號隔開 例如: k_2, k_3, k_4, k_5 表示貨櫃 k_2, k_3, k_4, k_5 可滿足條件，若有超過一組解，則輸出字典序較小的解答 (較早出現的)。

範例：

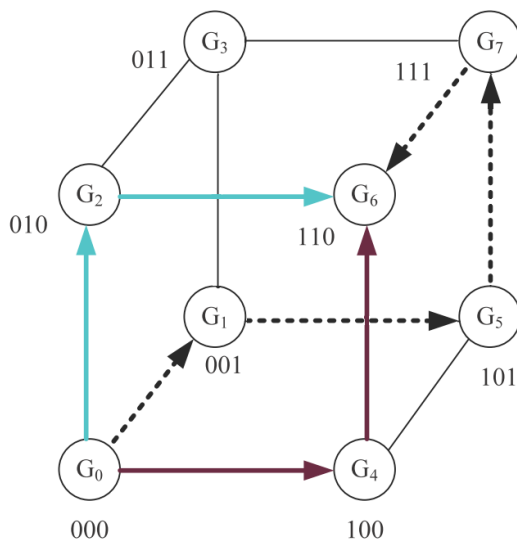
Sample Input:	Sample Output:
15,4,9,1,2,3,4,5,6,7,8,9	k_0, k_1, k_2, k_8
200,6,10,21,27,34,41,54,61,17,18,19,10	$k_0, k_1, k_2, k_3, k_5, k_8$

Problem 3. Disjoint path in hypercube

(Time Limit: 2 seconds)

問題描述：

Hypercube 又稱超立方體，因為有著優異的連接特性，可用來建構網路的拓樸及平行傳輸等相關領域。其中，不重疊路徑 (disjoint path) 是個很有趣的特點。在給定超立方體規模大小 (m) 的時候，該超立方體會由 2^m 個節點 ($0 \sim 2^m - 1$) 組成，當指定要連接某來源點 (s) 到某目標點 (d) 時，其必有 m 條不重疊的路徑。如下圖所示，



要找出規律並不難，所有的秘密皆在於二進制表示法裡面。

提示：由最左邊的位元開始處理。

輸入說明：

輸入包含若干筆測資，每筆測資包含超立方體的規模大小 (m)，來源點 (s) 和一目標點 (d)， $2 < m < 32$ ， $0 \leq s, d \leq 2^m - 1$ 。

輸出說明：

每筆測資請輸出所有不重疊的路徑，每一行一筆路徑。

範例：

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

	0 1 3
	0 2 3

Problem 4. 算數移位相加

(Time Limit:2 seconds)

問題描述：

某人在一長串的十進制 n 位元記憶體中做正整數數值的算術右移與算術相加運算。假設此 n 位元的原數值為 a_0 則執行 n 次運算後即可得到一系列數值 a_0, a_1, \dots, a_n ，然後將所有加總起來得到 $S = \sum_{i=0}^n a_i$ 。

如：

$a_0 = 1234$

$a_1 = 0123$

$a_2 = 0012$

$a_3 = 0001$

$a_4 = 0000$

相加總即可得到 $S=1370$ 。

假設現在只給予加總結果 S ，請你找出此 n 位元的原數值 a_0 ？。

輸入說明：

輸入若干個正整數，每行為一個測資，測資範圍為 $1 \leq S \leq 10^{n-1}$ ， $n = 19$ 。

輸出說明：

針對每一筆測資，如果有解即輸出 a_0 ，否則輸出 -1。

範例：

Sample Input:	Sample Output:
9999999999999999999	9000000000000000000
565	-1

Problem 5. Gray code generation

(Time Limit: 5 seconds)

Problem Description

A Gray code is an encoding of numbers so that two contiguous numbers have a single digit differing by 1. The term Gray code is often used to refer to a “reflected” code, or more specifically still, the binary reflected Gray code. For example, one-bit Gray code is $G_1 = (0, 1)$ and two-bit Gray code is:

G_2	00
	01
	11
	10

Three-bit Gray code is:

G_3	000
	001
	011
	010
	110
	111
	101
	100

This problem asks you to list the n -bit Gray code using two stacks. Your program should read a value n and use two stacks to help you to generate the n -bit Gray codes as well as list them in order. The listed order must be the same as the samples shown in this problem.

Input Format

There are several test cases. For each test case, the input is an integer, n , the number of digits for the Gray code, where $n \leq 15$.

Output Format

For each test case, the output contains n lines for each n-bit Gray code generated. Note that the order should be the same as all the samples shown in this problem.

Example

Sample Input:	Sample Output:
2	00
4	01
	11
	10
	0000
	0001
	0011
	0010
	0110
	0111
	0101
	0100
	1100
	1101
	1111
	1110
	1010
	1011
	1001
	1000