**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：1640: The Couinting Problem**

**解題者：林暐智**

**解題日期：2025年3月27日**

**題意：**

給定兩個正整數 a,b (0<a,b<100000000)，計算區間 [a, b] 中，0～9 分別出現的次數。

**題意範例：**

1 10 🡺 1 2 1 1 1 1 1 1 1 1

44 497 🡺 85 185 185 185 190 96 96 96 95 93

346 542 🡺 40 40 40 93 136 82 40 40 40 40

1199 1748 🡺 115 666 215 215 214 205 205 154 105 106

1496 1403 🡺 16 113 19 20 114 20 20 19 19 16

1004 503 🡺 107 105 100 101 101 197 200 200 200 200

1714 190 🡺 413 1133 503 503 503 502 502 417 402 412

1317 854 🡺 196 512 186 104 87 93 97 97 142 196

1976 494 🡺 398 1375 398 398 405 499 499 495 488 471

1001 1960 🡺 294 1256 296 296 296 296 287 286 286 247

0 0 🡺 結束程式

**解法：**

利用數位計數方法來找[1,n] 中，0～9 分別出現次數，最後將[1,b]-[1,a-1]=[a,b]即為所求。

(1)逐位掃描每一個數字（從個位數到最高位）

(2)將當前位拆成三個部分：

higher：比當前位高的數字

current：當前位的數字

lower：比當前位低的數字

如果目標數字 d>0 是當前位數的數字：

當 current < d：higher \* 權重

當 current == d：higher \* 權重 + lower + 1

當 current > d：(higher + 1) \* 權重

如果目標數字 d=0 是當前位數的數字：

當 higher ==0：跳出迴圈

當 current >0：higher \* 權重

當 current ==0：(higher - 1) \* 權重+ lower + 1

**解法範例：**

1 47 🡺 4 15 15 15 13 5 5 5 4 4

以d=1為例:

1.個位數(權重為1):  
higher=4、current=7、lower=0

因為current > 1 → (higher + 1) \* 權重=(4 + 1) \* 1 = 5

(在 0~9, 10~19, 20~29, 30~39, 40~49 這 5 個區間中各出現 1 次)

2.十位數(權重為10):

higher=0、current=4、lower=7

因為current > 1 → (higher + 1) \* 權重=(0 + 1) \* 10 = 10

(在10~19出現共10次)

所以[1,47]區間 ”1” 一共出現 5 + 10 = 15次

**討論：**

(1) 單純用暴力法會TLE，所以才採取”數位計數方法”，

其中，時間複雜度為O(log n)。

(2)當d>0與d=0要帶不同公式，因為當 high 為 0 時，這一位不產生「有效」的 0。

**程式：**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

// 計算從 1 到 n 中，數字 d 出現的次數

int countOccur(int n, int d) {

if(n < 1) return 0;

int count = 0;

// i 依次代表 1, 10, 100, (個位、十位、百位...)

for(int i = 1; i <= n; i \*= 10) {

int divider = i \* 10;

int high = n / divider; // 當前位左邊的部分

int current = (n / i) % 10; // 當前位數字

int low = n % i; // 當前位右邊的部分

if(d > 0) {

if(current > d)

count += (high + 1) \* i;

else if(current == d)

count += high \* i + low + 1;

else

count += high \* i;

}

else { // d == 0 要特殊處理，因為前導 0 不算

// 當 high 為 0 時，這一位不產生「有效」的 0（因為 n 的位數不足）

if(high == 0) break;

if(current > 0)

count += high \* i;

else

count += (high - 1) \* i + low + 1;

}

}

return count;

}

// 計算區間 [a, b] 中，0～9 各出現的次數

vector<int> countRange(int a, int b) {

if(a > b) swap(a, b);

vector<int> result(10, 0);

for (int d = 0; d < 10; d++) {

result[d] = countOccur(b, d) - countOccur(a - 1, d);

}

return result;

}

int main(){

ios::sync\_with\_stdio(false);

cin.tie(nullptr);

int a, b;

while(cin >> a >> b && (a || b)) {

vector<int> ans = countRange(a, b);

for (int i = 0; i < 10; i++){

cout << ans[i] << (i < 9 ? " " : "\n");

}

}

return 0;

}