



# 全國高級中等學校 97 學年度商業類科學生技藝競賽

## 【 程式設計 】 職種【術科】試卷(模擬試題)

選手證號碼：\_\_\_\_\_ 姓名：\_\_\_\_\_

### Problem1：字數統計(14%)

雷迪雅被老師要求寫一份為數 2000 字的報告，可是雷迪雅胸無點墨、才疏學淺，常常寫了一整個下午還不到一頁，因此他每寫一段時間就開始數自己已經寫了多少個字，可是常常這樣數也是會累的，於是他希望能有一個自動字數統計程式。

有了字數統計程式，就可以自動統計好一篇文章總共有幾行、有幾個字、有幾個字元。行被定義為用換行字元隔開的連續字元，字被定義為用空白、TAB 或換行字元所隔開的連續字元，而字元除了一般可見的字元還包括 TAB 字元和空白字元(注意不包含換行字元)。

身為雷迪雅好朋友的你，常常受到他的照顧，正所謂吃人嘴軟，拿人手短，如今雷迪雅遇到了這個難題，請你義不容辭地寫一個程式幫他吧！

### 輸入說明：

輸入檔第一行說明有幾組測試資料，第二行開始即為第一筆測試資料，每行不會超過 1024 個字元，每組測試資料中間用五個連續等號 '=' 的一行來作分隔。每組測試資料之中絕不會有五個連續等號 '=' 出現。

### 輸出說明：

每組測試資料輸出一行，每行有三個數字，分別代表一組測試資料中有幾行，幾個字和幾個字元，每個數字之間請用一個空白隔開。

### 輸入範例：

```
2
This is a sample input.
Hello World!!
=====
The speech by Hunyak, translated, is:
"What am I doing here?
They say, the famous Hungarian police,
that I held down my husband and chopped off his head.

But I didn't do it, I am not guilty.
I can't believe that Uncle Sam says I did it.
They say I did it, but really I didn't."
```

### 輸出範例：

```
2 7 36
8 55 270
```

**Problem2：惱人的零錢(15%)**

東東在買東西付帳，總是習慣直接從錢包中拿鈔票付帳，而懶得掏出硬幣來。久而久之，錢包裡面累積了許多硬幣，簡直重得不得了，所以東東終於受夠了！因此，她決定趁著今天買東西的時候，想辦法盡量減輕負擔。於是東東開始盤算要怎樣湊出足夠的硬幣，才能讓付出去的硬幣個數越多越好。同時，目前這家店的老闆人很好，不論客人給他多少硬幣，他都一定會用最少的硬幣找錢給客人。所以，當東東走到櫃台結帳時，東東想到自己如果多付一些硬幣讓老闆找錢，說不定可以讓自己的錢包更輕！因此，東東開始煩惱到底要怎麼給錢，才能夠盡量「用掉」最多的硬幣呢（所謂的「用掉」的硬幣個數，指的是拿出去的硬幣數，扣掉老闆找回來的硬幣數）？可惜的是，東東的算術一向不太靈光，因此希望你能幫忙他解決這個煩惱。

**輸入說明：**

輸入資料的第一行是一個整數  $n$ ，代表共有  $n$  筆測試資料。接下來每筆測試資料有 3 行：第 1 行的數字  $C$  表示要買的東西的價格。第 2 行有 5 個數字  $p1\ p5\ p10\ p20\ p50$ ，分別是東東錢包裡面一元、五元、十元、二十元和五十元硬幣的個數。第 3 行有 5 個數字  $q1\ q5\ q10\ q20\ q50$ ，是老闆所擁有的一元、五元、十元、二十元和五十元硬幣的個數。每筆測試資料的所有數字都在 0 到 10000 之間；同一行的數字之間會用一個空白隔開。你可以假設東東身上的錢足夠來購買該商品，而且至少有一種付錢的方法使得老闆可以找得開（如果需要找錢的話）。因為老闆和東東很不幸地很碰巧地一張鈔票都沒有，請不要問說為什麼不能換成大鈔。

**輸出說明：**

你的輸出資料應該有  $n$  行，分別對應到  $n$  筆輸入的測試資料。每一行要輸出一個數字表示東東付完帳之後，剩餘的硬幣總數。

**輸入範例：**

```
2
25
10 3 2 1 3
0 0 0 0 0
25
0 3 2 2 3
1 1 1 1 1
```

**輸出範例：**

```
6
4
```

**Problem3：基礎排序(13%)**

給一堆數字，把他們從小到大排序好。

**輸入說明：**

每組測試資料共有兩行，第一行的數字  $n$  為有幾個數字要排序，第二行則有  $n$  個整數( $n \leq 1000$ )，其餘整數皆於-10000 到 10000 之間，測試資料中包含多組測試，當排序個數為 0 時結束。

**輸出說明：**

輸出已排序好的數列，每個數字之間請用一個空白隔開。

**輸入範例：**

5  
5 4 3 2 1  
5  
-1 -2 -3 -4 -5  
0

**輸出範例：**

1 2 3 4 5  
-5 -4 -3 -2 -1

**Porblem4：百貨公司打折程式(12%)**

豪慷慨百貨公司週年慶的打折策略，吸引了許多客人上門，因此公司決定再回饋客戶，當客戶消費超過 2000 元時打 7 折，消費超過 5000 元時打 6 折，消費超過 10000 元時打 55 折。請幫該公司寫出一個新的收銀台程式，輸入顧客購買總金額  $n$  後，計算顧客實際需付的錢。

**輸入說明：**

購買金額  $n$

**輸出說明：**

實付金額

**輸入範例：**

3000  
6000  
12000

**輸出範例：**

2100  
3600  
6600

**Problem5：糊塗情報員(16%)**

有一位間諜，依他所屬情報單位要求編碼的方式，將他所收集到情報全部編成數字碼。但他認為這樣還是不夠安全，因此他再將這些數字字串，隨意切割成好幾個整數，然後將每個整數用一個數學算式來表示。這些算式只用了加、減、乘三種運算子，而且每個運算元都是正整數。最後，他為了讓他自己更為心安，他將整個密碼分成兩本密碼簿儲存。密碼本 A 存放這些數學算式，但他將算式內的所有括號全部拿掉，然後再將這些拿掉的括號資訊記錄在密碼本 B 裡面。

過了不久，這個間諜發現他把密碼本 B 弄丟了，再加上他的記憶力不好，很多情報內容根本記不得，所以現在沒了密碼本 B 幾乎束手無策。在不得已的情況下，他的情報單位派了幾位心理與腦神經專家詢問他，希望能喚起他腦海內的記憶。這些專家試了好幾天，用盡各種辦法後，終於承認他的記憶力果真很差，怎麼也問不出情報內容。倒是心理學專家有一發現，即這位情報員在寫密碼算式時，傾向於將括號加在那些會讓算式得最大值的位置。例如  $5*7+2$  這個算式，有兩種括法： $((5*7)+2)$ 以及 $(5*(7+2))$ ，第二種括法所得的值較大。請寫一程式，算出這些算式的可能最大值。

**輸入說明：**

每一筆輸入資料為一行算式，運算子只有三種，即一般的加、減、乘三種二元運算子，分別以符號 '+' 、 '-' 、 '\*' 表示。每一個運算元都是一個正整數( $\leq 100$ )，運算元和運算子之間不會有空白，一行算式不會有超過 50 個運算元。

**輸出說明：**

相對於每一輸入算式，輸出所有可能運算結果的最大值。該值都會是一個正整數，而且不會超過 2147483647。

**輸入範例：**

5\*7+2  
6\*3-9\*3  
5+2-7\*2-3

**範例輸出：**

45  
27  
14

**Problem6：中文大寫數字(13%)**

我們在金融機構填寫金額時使用的不是阿拉伯數字，而是中文的大寫數字。請寫一個程式將數字轉換為中文大寫數字。

標準大寫寫法如下：零、壹、貳、參、肆、伍、陸、柒、捌、玖、拾、佰、仟、萬、億

**輸入說明：**

整數數字  $n$  ( $0 \leq n \leq 2147483647$ )。

**輸出說明：**

文字字串，遇到 10 時，請輸出『壹拾』。

**輸入範例：**

12345  
10200

**輸出範例：**

壹萬貳仟參佰肆拾伍  
壹萬零貳佰

**Problem7：奧步戰術(17%)**

在黑暗算法界中，使用奧步解題似乎已經漸漸成為主流。雖然使用奧步將漸漸使人走向魔路，最後被內心的虛無吞噬，不過這不是今天的問題。考慮在某個考試中，有  $n$  道題目，而總答題時間為  $T$ 。對於每題都只有三種可能：

- 1. 正解能得到全對的分數(得 2 分)
- 2. 奧步能拿到半對(得 1 分)
- 3. 放棄的話當然就沒分囉(0 分)

而對每題來說，要達到這三種分數所需花的時間皆不同，所有題目拿 0 分都不用花費時間；在題目  $i$  使用奧步拿半對所需時間為  $Hi$ ，要寫正解所需時間為  $Ci$ ，其中對於任何題目  $i$ ，必有滿足  $0 < Hi < Ci$ 。

試問：在時間  $T$  內，用最佳的答題方式，最多可以拿幾分？

**輸入說明：**

輸入檔第一行說明有幾組測試資料，第二行有兩個整數  $n$  和  $T$ ，分別代表有幾題，以及總作答時間。接下來  $n$  行每行有兩個整數  $Ci$  和  $Hi$ ，代表第  $i$  題寫正解需要時間  $Ci$ ，寫奧步需要時間  $Hi$ 。其中：

- 題目總數  $n \leq 100000$
- 答題所需時間  $1 \leq Hi, Ci \leq 1000000$
- 總作答時間  $0 < T \leq 1000000000$

**輸出說明：**

每個測試範例請輸出一個整數，代表最大得分。

**輸入範例：**

2  
5 12  
4 3  
6 2  
5 3  
4 3  
5 2  
4 10  
5 3  
6 5  
3 1  
4 3

**輸出範例：**

6  
5