

臺北市立第一女子高級中學 103 學年度

資訊學科能力競賽初賽



程式設計試題

2014.06.10

答題注意事項：

1. 請留意題目的說明，題目中未提及的事項，請勿自行假設。例如題目若要求輸入一個整數，則勿假設此整數必為正整數。
2. 評分時的測試資料由評分老師提供，僅測試完成題目上的輸入輸出範例，並不能保證該題能得到滿分。
3. 每題程式需在 10 秒內執行出結果，否則不予計分。
4. 輸出格式需完全和題目規定相同，不可以輸出多餘空白。
5. 不可以加 `system("pause");` 等系統函數，如因這樣而無法 judge，請自行負責。
6. 程式編寫時間：09:10 ~ 11:40
7. 程式測試時間：11:45 ~ 12:10

Problem A：影像二值化



Figure A-1
Color Image



Figure A-2
Gray scale Image



Figure A-3
Binary Image(t = 50)



Figure A-4
Binary Image(t = 30)

彩色影像檔的每個像素 (pixel) 包含紅(R)、綠(G)、藍(B)三個顏色的亮度值。灰階 (Gray scale) 影像檔的每個像素為黑色(0)到的白色(255)不同亮度的階度值。人眼對綠色的亮度感最大，而對藍色最小，在心理學的角度來看，較適合人類眼睛的灰階影像方程式如下：

$$\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B$$

影像二值化 (Binary Image) 在圖形識別裡佔了很重要的地位。它是一個將影像以 0 與 1 表示的影像處理過程。首先，將彩色影像的每一像素轉換成灰階值 G 後，再與門檻值 (threshold, t) 相比：若計算得到的灰階值 G 小於該門檻值 t，則將此點的顏色設為黑色(0)；否則，將此點的顏色設為白色(1)。

註：此計算過程中所得的灰階值 G 為一浮點數，僅為與門檻值 t 進行比較，毋需轉換成整數值。

門檻值 (t) 的取得有很多方法，我們採取一個最簡單最快速的方式，將所有像素的灰階值 G 加總後取平均。

Task

給定一個 $n \times n$ 的彩色影像檔，內含每一像素的 RGB 值。請將此彩色影像轉換成 Binary Image，輸出門檻值 t、並計算白色像素個數與黑色像素個數。

Input

第一行為整數 n ($0 < n \leq 128$)，表示此影像為 $n \times n$ 大小。之後有 n 行，每一行代表 n 個像素，每一個像素皆由 RGB 三個顏色值組成(每個顏色的值是介於 0 到 255 之間的整數，整數間以一個空白字元分隔)。

Output

共一行，有 3 個以空白隔開的數值。其中，t 為門檻值(顯示至小數點後第三位)、w 為白色像素個數、b 為黑色像素個數(如 Sample Output 所示)。

[註 1]：C++ 語言控制顯示小數位數的方法
可使用 `setprecision` 函式，但需 `#include <iomanip>` 函式庫。
例如：`cout << fixed << setprecision(2) << 3.1234 << endl;`
顯示的結果為 3.12

[註 2]：C 語言控制顯示小數位數的方法
`%.2f` — 印出小數點後兩位的浮點數。
例如：`printf("%.2f", 3.1234);`
顯示的結果為 3.12

Sample Input

```
5
14 72 55 67 135 177 194 144 15 9 40 84 37 228 128
216 58 0 91 246 19 30 75 137 110 16 166 34 232 20
135 253 30 96 22 51 19 254 9 48 210 218 107 104 84
26 156 70 163 173 154 101 64 39 126 3 194 204 74 179
193 234 225 81 242 109 87 59 147 173 107 190 10 138 82
```

Sample Output

```
118.531 13 12
```

Problem B：小小音樂會

小綠綠要舉辦期末小小音樂會，音樂老師希望同學們依照演奏樂曲的時間長短來安排演出的順序。上、下半場時間大約相同。若無法平分，上半場時間需大於下半場時間。

在音樂理論中，速度（tempo）決定了一段音樂的快慢，是音樂的重要元素，亦影響作品的情感與演奏難度。音樂速度一般以文字或數字標記於一首樂曲的開端，現代習慣以每分鐘多少拍（beats per minute，簡稱BPM）作單位。演奏速度「♩=120」表示：120 是其 BPM 值，表示每分鐘演奏 120 個四分音符[♩]。

舉例來說：4/4 拍 27 小節的曲子，速度為「♩=60」，問其演奏時間？

4/4 拍意思是每 4 個四分音符 [♩] 成 1 小節，共 27 小節，則全曲有 $27 \times 4 = 108$ 個 [♩]。「♩=60」表示每分鐘演奏 60 個四分音符 [♩]，所以 108 個共需 $108/60 = 1$ 分又 48 秒。

再舉一例來說：6/8 拍曲子共 150 小節，速度為「♩.= 120」，問其演奏時間？

6/8 拍意思是每 6 個八分音符 [♩.] 成一小節。共 150 個小節，表示全曲共有 $150 \times 6 = 900$ 個 [♩.]。附點四分音符 [♩.] 的時值是三個 [♩] 那麼長，所以 900 個 [♩] 的長度就等於 $900 / 3 = 300$ 個 [♩.]。「♩.= 120」的意思就是每分鐘會演奏掉 120 個 [♩.]，所以 300 個 [♩.] 共需 $300 / 120 = 2.5$ 分鐘，也就是 2 分 30 秒演奏完。

Task

現在，請為小綠綠寫出一套自動安排節目表的程式。計算每首曲目的演出時間，計算總演出時間，依序排序後，分出上半場與下半場的演出。

Input

第一行為演出曲目數 n ($0 < n < 50$)。

接下來的 n 行，每一行有演出曲目名稱、節拍、總小節數、演奏速度。

例如：A 曲目為 4/4 拍 27 小節的曲子，演奏速度為「♩=60」記錄如下：A [4 4] 27 [60 1]。B 曲目為 6/8 拍 150 小節的曲子，演奏速度為「♩.= 120」記錄如下：B [6 8] 150 [120 1.5]。

Output

第一行為總演出時間 (m' s")。

m 表示分鐘，s 表示秒數（四捨五入並顯示至整數位）。

接下來有 $n+1$ 行，依演奏時間由小至大排序。

若演奏時間相同時，以曲目名稱之字典序排序。

每一行包含曲目名稱與演奏時間 (m' s")，以空白隔開。

m 表示分鐘，s 表示秒數（四捨五入並顯示至整數位）。
上下半場中間以三個減號「---」隔開，如 Sample Output 所示。

[註 1]：C++ 語言控制顯示小數位數的方法

可使用 `setprecision` 函式，但需 `#include <iomanip>` 函式庫。
例如：`cout << fixed << setprecision(2) << 3.1234 << endl;`
顯示的結果為 3.12

[註 2]：C 語言控制顯示小數位數的方法

`%.2f` — 印出小數點後兩位的浮點數。
例如：`printf("%.2f", 3.1234);`
顯示的結果為 3.12

[註 3]：四捨五入的方法

以四捨五入至小數**第二位**為例，處理步驟如下：

1. 原數 x 乘以 100，再加上 0.5，再將結果強制轉換為整數 `int` 型別，即可無條件捨去小數的部份。
2. 由於之前乘以 100 放大，所以最後要再除以 100 還原。
請注意除數 100.，在 100 之右要加小數點，使這個除式得以浮點運算進行。

```
double xx;  
xx = (int)(x * 100 + 0.5) / 100.;
```

若你要四捨五入至小數第三位，把上述的 100 改成 1000 即可，餘類推。

Sample Input

```
3  
AA 4 4 27 60 1  
BB 6 8 150 120 1.5  
CC 3 2 10 72 1
```

Sample Output

```
5' 8"  
CC 0' 50"  
AA 1' 48"  
---  
BB 2' 30"
```

Problem C：變動長度編碼 (Run-Length Encoding, RLE)

變動長度編碼 (run-length encoding) 是一種簡單的非破壞性資料壓縮法，壓縮和解壓縮都非常快。它能夠用來壓縮由任何符號所組合而成的資料。

這個方法是利用一個符號加上一個數字來取代連續出現的相同符號。例如，‘AABBBBDAA’就以 A2B4D1A2 來取代。原始字串有 9 個字元，壓縮後的字串有 8 個字元，壓縮率為 $8/9 = 0.889$ 。變動長度編碼的優點是編碼快速，但缺點是對於不重覆的資料反而加大容量。

Task

請將一個字串資料進行 run-length encoding，並計算該字串的壓縮率。

Input

第一行是一個整數 N ($N < 50$)。代表測試筆數。

接下來 N 行，每一行有一組字串，字串長度不超過 1000，每一連續字元長度不超過 9，如 Sample Input 所示。

Output

輸出 N 行。

每一行有 2 個資料 S, R ，兩者以半形空白隔開。

其中 S 為編碼過的字串， R 為該字串的壓縮率 (顯示至小數點後三位)。

如 Sample Output 所示。

Sample Input

```
3
AABBBBDAA
AAAAAAAAABBBB
ABCD
```

Sample Output

```
A2B4D1A2 0.889
A8B4 0.333
A1B1C1D1 2.000
```

[註 1]：C++ 語言控制顯示小數位數的方法

可使用 `setprecision` 函式，但需 `#include <iomanip>` 函式庫。

例如：`cout << fixed << setprecision(2) << 3.1234 << endl;`

顯示的結果為 3.12

[註 2]：C 語言控制顯示小數位數的方法

`%.2f` — 印出小數點後兩位的浮點數。

例如：`printf("%.2f", 3.1234);`

顯示的結果為 3.12

Problem D：自動對獎



單數月的 25 日，對小綠來說，是一個既期待，又怕傷害的日子。因為收集了兩個月的一大疊發票終於可以拿來兌獎了。然而最令小綠心煩的，就是「對發票」的過程。一張一張的確認，不僅耗時間耗精力，更可能發生漏對的情況。

好友 HS 幫小綠設計了一個自動發票掃描器，可以將一整疊的發票號碼逐一掃描進入，省去了看發票的辛苦，小綠覺得 HS 真是太貼心了~~ CK 知道了，也不甘示弱，他要設法讓小綠可以自動對發票。可是，CK 在程式設計課時都沒專心聽講，身為 CK 的姐姐，妳可以幫幫他嗎？

統一發票對獎規則如下：

特別獎	同期統一發票收執聯 8 位數號碼與特別獎中獎號碼相同者獎金 1,000 萬元
特獎	同期統一發票收執聯 8 位數號碼與特獎中獎號碼相同者獎金 200 萬元
頭獎	同期統一發票收執聯 8 位數號碼與頭獎中獎號碼相同者獎金 20 萬元
二獎	同期統一發票收執聯末 7 位數號碼與頭獎中獎號碼末 7 位相同者各得獎金 4 萬元
三獎	同期統一發票收執聯末 6 位數號碼與頭獎中獎號碼末 6 位相同者各得獎金 1 萬元
四獎	同期統一發票收執聯末 5 位數號碼與頭獎中獎號碼末 5 位相同者各得獎金 4 千元
五獎	同期統一發票收執聯末 4 位數號碼與頭獎中獎號碼末 4 位相同者各得獎金 1 千元
六獎	同期統一發票收執聯末 3 位數號碼與頭獎中獎號碼末 3 位相同者各得獎金 2 百元

Input

前 5 行為中獎號碼。第 1 行為特別獎、第 2 行為特獎、第 3-5 行為頭獎。

第 6 行為一個正整數 N，代表小綠的發票張數。

接下來 N 行。為小綠的發票號碼，每一組號碼為 8 碼。

Output

若全無中獎，輸出「Thanks for patronage」

若有中獎，輸出中獎總金額。

Sample Input 1

34231966
57287066
29828540
85564445
39775689
5
47696540
38932438
39335689
57287065
43231966

Sample Output1

1200

Sample Input 2

34231966
57287066
29828540
85564445
39775689
5
47696541
38932438
39335688
57287065
43231966

Sample Output2

Thanks for patronage

Problem E：秘密紙條

湯姆和貝琪是一對兩小無猜的好朋友。他們喜歡在上課時傳紙條，又怕被同學攔截，大聲的唸出來。湯姆想到在電腦課上學到二進位的計算方式與ASCII的編碼，他發現，如果把要給貝琪的訊息轉成8位元的ASCII碼，再以0與1的二進位形式寫出來，同學就不會知道他們傳遞的訊息是什麼了。



例如：湯姆想傳「I miss you.」由ASCII轉換後的binary編碼如下：

Message	I		m	i	s	s		y	o	u	.
ASCII	73	32	109	105	115	115	32	121	111	117	46
Binary	01001001	00100000	01101101	01101001	01110011	01110011	00100000	01111001	01101111	01110101	00101110

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

Figure E-1 ASCII Code

Task：請幫忙貝琪把湯姆傳來的訊息一一解碼。

Input：第一行為正整數 N ($0 < N < 100$)，代表訊息則數。接下來 N 行，每一行為連續 0 與 1 組合的字串，表示一則訊息。

Output：有 N 行。每一行為解碼後的訊息。

Sample Input

```
4
010001000111001001100101011000010110110101110011
01000011011011110110110101100101
01010100011100100111010101100101
0100100100100000011001000110111100101110
```

Sample Output

```
Dreams
Come
True
I do.
```