

## TFcis13 上學期留社考 程式設計能力上機測驗

命題範圍：基本、條件、迴圈、陣列、函式、遞迴、結構、排序、搜尋、字串

預定考試日期：99 年 12 月 29 日

考試注意事項：

1. 本測驗時間共 2 小時 30 分鐘。
2. 本測驗直接在電腦上作答，考試時間結束後須繳交原始碼檔案，題目卷則不收回。(有需要可在題目卷上做計算)
3. 若題目有敘述不清的部分，請於考試開始後 60 分鐘內提問。超過 60 分鐘後，不再回答與題目相關的問題。
4. 每題可只答對部分測試資料。
5. 每題均有執行時間限制，超過時限該測資以 0 分計。
6. 測試程式以原始碼為主，即使已編譯成可執行檔，測試時仍會重新編譯。
7. 輸出的最後一個字元，若無特殊提示，皆為換行字元「\n」。
8. 本次考試限用 C 或 C++ 語言。
9. 原始碼檔案名稱按照題號依序為 PRO1、PRO2、PRO3、……，C 原始碼副檔名為 c，C++ 則為 cpp。
10. C 編譯方式：gcc -O2 -Wall -ansi
11. C++ 編譯方式：g++ -O2 -Wall -static
12. 編譯器：MinGW GCC 4.5.0-1

扣分規則表：

該題 0 分	使用作業系統特有的 API 或任何非標準函式 (例： <code>strcmp()</code> 、 <code>_findfirst()</code> )
該題 0 分	程式對檔案進行、寫入的操作 (使用檔案操作函式，但沒有真正操作檔案則不限制)
該題 0 分	編譯錯誤 (Compilation Error, CE)
該題扣 1 分	檔名未依規定
該題扣 1 分	因結尾有 <code>system("pause")</code> 、 <code>while(1)</code> 或其他項目而導致程式執行完未結束
該題扣 1 分	程式正常結束，卻未回傳 0 (可能是 <code>main()</code> 未加上 <code>return 0</code> )
該測資得分減半	因缺少或多出空格、換行，而導致格式錯誤 (即 Presentation Error, PE)

## 第 1 題：功能強大的複製程式

執行時限：3 秒 配分：10 分

因為檔案總管不太好用，又不會自己比較檔案，所以你決定找個好用的檔案複製程式。

最近你從 T12 網管教學(cp123cp123=cpy)那裡拿到一個功能超強的複製程式，叫做 cp321，支援檔案內容、日期比對及複製，並且會留下紀錄檔，方便日後查閱相關訊息。不過，這個程式所留下的訊息非常簡單，而且複製過程中也沒有任何訊息出現，所以也只能靠查閱紀錄檔了解相關資訊。然而，因為最近在做資料備份，紀錄檔太多，所以需要請你寫一個程式來分析這些紀錄檔。每次複製都會在紀錄檔加入若干行資料，一個紀錄檔中可能有多筆紀錄。所以你必須以「版權宣告」來判斷是否已經進入下一次的紀錄。

每筆紀錄都有 5 個部份，以下列出：

第 1 部分	cp321 版本資訊以及版權宣告，一定只有 1 行。 每次可能都不一樣，但保證行首一定是 cp321，後面接一個空格。
第 2 部分	要求的操作名稱，可能不只一行。裡面一定會包含要處理的檔案名稱，檔案名稱以「`」和「'」圍住。可以保證在其他地方不會出現「`」和「'」符號。
第 3 部分	複製開始的時間，僅有 1 行。行首會有 BEGIN: 提示。 BEGIN: 後方有一個空格。
第 4 部分	複製結束的時間，僅有 1 行。行首會有 END: 提示。 END: 後方有一個空格。
第 5 部分	檔案總大小 (位元組)，僅有 1 行。行首會有 SIZE: 提示。 SIZE: 後方有一個空格。

但是，時間都是以一種很難辨識的方式表示，也就是距離 1970 年 1 月 1 日 0 時 0 分 0 秒的秒數。不過還好，你只需要計算時間差值就可以了。

## 輸入

輸入 1 個紀錄檔，裡面有多次複製所留下的紀錄資料。每次紀錄間不會有任何空行或分隔符號。**每行最多 255 個字元**。

時間一定可用 32-bit unsigned integer 儲存，檔案大小一定可以用 64-bit signed integer 儲存。

## 輸出

針對記錄檔中的每筆紀錄，依序輸出檔案名稱、複製所花的時間，檔案總大小。其中時間必須以「0:57:24」這種時、分、秒分離的方式表示（分、秒未達兩位數要補 0），檔案大小則要視情況加上 B、KB、MB、GB，使數字小於 1024，且四捨五入到整數。

但是，如果檔案大小已經以 GB 表示，數字仍在 1024 以上，則不限制。（請參考範例輸出）

## 範例輸入

```
cp321 Version 4.4 2010-12-24 (C) All right reserved.  
Copying directory `foo' ...  
BEGIN: 175548889  
END: 175548900  
SIZE: 53544  
cp321 Version 3.57 2007-02-24 (C) All right reserved.  
Initializing and establishing link...  
Client IP is 192.168.3.89  
Detecting remote operating system... FreeBSD 7.0  
Copying file `init' ...  
BEGIN: 1755245556  
END: 1755300007  
SIZE: 98756542315  
cp321 Version 2.87 2004-05-01 Created by sda  
Comparing files...  
Copying file `loadfix.exe'  
BEGIN: 3257789665  
END: 3257792573  
SIZE: 66875920  
cp321 Version 1.0 1999-08-24 Created by sda  
Copying file `reset.tar.bz2'  
BEGIN: 57894562  
END: 60000000  
SIZE: 78789789798  
cp321 Version 4.41 2010-12-29 (C) All right reserved.  
Preparing...  
Creating `very big file'  
BEGIN: 25278  
END: 47486  
SIZE: 324789074983278
```

## 範例輸出

```
foo: 0:00:11 52 KB
init: 15:07:31 92 GB
loadfix.exe: 0:48:28 64 MB
reset.tar.bz2: 584:50:38 73 GB
very big file: 6:10:08 302483 GB
```

## 測資說明

其中 20%的測試資料，紀錄檔中只有一筆紀錄。

另外 80%的測試資料，紀錄檔中有多筆紀錄。

## 提示

1. 包圍在檔案名稱左側的「`」符號位於鍵盤左上方  
包圍在檔案名稱左側的「'」符號位於鍵盤 Enter 鍵旁邊
2. 以範例輸入為例：  
1~5 行是第 1 筆紀錄  
6~13 行是第 2 筆紀錄  
14~19 行是第 3 筆紀錄  
注意檔案結尾還有一個空格

## 第 2 題：Programming Style

執行時限：10 秒 配分：15 分

最近 T12 網頁教學(tony2guo)忙著製作社內網站([www.tfcis.org](http://www.tfcis.org))，因此常在不同的電腦上寫網頁，由於每個編輯器的縮排方式和 Tab 字元設定都不太一樣，因此需要一個轉換程式來處理這些問題。尤其是 Tab 字元常會用來縮排，但不同編輯器的設定卻不相同。如果不用 Tab 字元而以空白字元縮排，雖可以避免問題，卻會使檔案增大許多，這對於開放下載的文件其實是不太方便的。

tony2guo 最近沒時間寫個程式來進行 Tab 字元和空白字元的轉換，所以想找高二的程設組社員來幫他寫這個程式，但恰好高二社員都不在，所以想請你幫忙。

他後來又發現在 UNIX-like 的作業系統上，有個好用的工具叫做 wc，這個工具會顯示文字檔的行數、英文字數、總字元數。但其實英文字的個數並不重要，他想改成空白字元個數，因此你要在程式中加入計算行數、空白字元數、總字元數的功能。

### 輸入

要求的程式操作方式如下：

第 1 行以兩個字串（以空白分隔）代表要做的事。以下列出：

t2b	將 Tab 字元轉為空白字元，後面接一個正整數表示每個 Tab 字元要轉為幾個空白字元。
b2t	將空白字元轉為 Tab 字元，後面接一個正整數表示每幾個空白字元要轉為一個 Tab 字元（若空白字元未達規定個數則不轉換）。
wc only	不需轉換

只有行首的 Tab 字元或空白字元要轉換，Tab 字元不一定在行首

接下來一直到 EOF 則是要轉換的檔案內容。

### 輸出

如果沒有轉換，只須輸出一行表示行數、空白字元數、總字元數（含換行字元，包含檔案結束的那一個）三個數字，並以空白分隔。

如果有進行轉換，請在最後以兩行分別輸出轉換前和轉換後的行數、空白字元數、總字元數。請參考範例測資。

以下範例輸入輸出，若遇 Tab 字元會以<Tab 字元>表示，這只是為了區分，寫程式時仍然要以類似 `printf("\t")` 的方式輸出。

## 範例輸入 1

```
wc only
<Tab 字元>printf("Hello World!\n");
<Tab 字元>return 0;
```

## 範例輸出 1

```
2 2 38
```

## 範例輸入 2

```
t2b 4
<Tab 字元>signal(SIGBREAK, f1);
<Tab 字元> signal(SIGBREAK, f1); /* Control-Break */
<Tab 字元><Tab 字元>return 0;
}
```

## 範例輸出 2

```
signal(SIGBREAK, f1);
    signal(SIGBREAK, f1); /* Control-Break */
        return 0;
}
4 5 79
4 21 91
```

## 範例輸入 3

```
b2t 3
<Tab 字元>char a;
    while(1)printf("\n%d\n\n", a=getche());}
```

## 範例輸出 3

```
<Tab 字元><Tab 字元>char a;
<Tab 字元> while(1)printf("\n%d\n\n", a=getche());}
2 10 58
2 4 54
```

## 測資說明

其中 30% 的測試資料，第一行都是 wc only。

另外 30% 的測試資料，第一行都是 t2b。

最後 40% 的測試資料，第一行是 b2t。

## 提示

上面只有<Tab 字元>不是真正的字元，測資裡面會是真正的 Tab 字元。同時也請注意，上面的\n 真的是兩個字元「\」和「n」。

## 第 3 題：SB 數列

執行時限：10 秒 配分：20 分

因為 TFcis 12 上學期留社考的第 3 題是「SA 數列」，所以 TFcis 13 上學期留考第 3 題是「SB 數列」。(?)

社長(sb283)最近因為常常做 ACM 的題目，所以當遇到難題一直想不出來時，他就會 sa072686 的個版上尋求提示。又因為學校要求，每個禮拜都要看 SC(Studio Classroom · 空中英語教室)。很顯然這兩件是剛好和他的帳號形成一個只有三項的「等差字串列」(?)。即 SA、SB、SC。此時恰好學校也教到二下 2-6 遞迴數列，因此他想寫個程式來幫他計算某個遞迴數列的第 n 項。

由於社長今天偷懶，不想自己寫程式，於是請 charlie 幫忙。不過卻發現這個程式有些不太正常的地方，也就是執行後即使是在等待輸入，一樣會導致 CPU 使用率達到 100%，並且不斷項系統索取記憶體，直到可用記憶體剩下 63MB。

於是，sb283 打算找你幫忙，他不想用這個不知道被 charlie 動過什麼手腳的程式，幾乎導致電腦當機，且 Ctrl+Break 和 Ctrl+C 都無法結束程式。不過，這個程式的輸入要求還滿麻煩的，請仔細看好！

sb283 喜歡二階遞迴數列，也就是長得像  $a_n = pa_{n-1} + qa_{n-2} + r$  的遞迴式，其中  $p$ 、 $q$ 、 $r$  是常數，但有時因為數字會太大，只需輸出除以 8247596325 後的餘數即可。還有，因為不同的需求，有時需要作進位制轉換，像是轉成 2、8、16 進位的數字，記得要考慮每一種狀況！

他最近在看 SC 時，看到了在某一期的 Scrapbook 中的 Riddle。我已不記得完整的英文敘述，所以部分用中文來寫：

Q：哪一個數字最可怕？

A：7. Because 7, 8, 9.

看懂了嗎？seven, eight, nine 就是 seven ate nine. 也就是說，遇到特殊數字要以字串來表達，所以這題的輸出也要非常注意。

## 輸入

每行有 7 個字串 ( 以空白分隔 )，每個測資檔以 EOF 作為結尾。每個字串都以「項目=值」的方式表示。

以下列出可能的項目和值：

項目	值
p	即上面遞迴式的 p，是個正整數
q	即上面遞迴式的 q，是個正整數
r	即上面遞迴式的 r，是個整數
a1	此數列的第 1 項，是個整數
a2	此數列的第 2 項，是個整數 ( $a2 \geq a1$ )
s	dec 表示以 10 進位表示 hex 表示以 16 進位表示 ( 英文字母部分要大寫 ) oct 表示以 8 進位表示 bin 表示以 2 進位表示
n	要求出的項目

以上的項目沒有一定的順序，可以任意交換。

以上 p、q、r、a1、a2 都可用 16-bit unsigned integer 儲存。

## 輸出

針對每一行，以規定的格式輸出數列第 n 項除以 8247596325 的餘數，不過，如果答案恰好等於以下幾項，則不輸出數字，以規定的字串代替。

特殊數字	規定的字串
7	This is a bad number.
8	This is a transitive verb.
9	This number has been deleted.

## 範例輸入 1

```
p=1 q=1 r=0 n=4 a1=1 a2=2 s=dec
p=1 r=0 n=7 q=1 a2=1 s=hex a1=1
p=2 q=4 a1=12 a2=87 r=13 n=4 s=oct
s=bin p=3 n=8 r=0 q=5 a1=0 a2=2
```

## 範例輸出 1

```
5
D
1477
10000100001010
```

## 範例輸入 2

```
p=1 q=1 r=1 n=5 a1=1 a2=1 s=bin
p=1 q=1 r=0 n=6 a1=1 a2=1 s=hex
p=1 q=1 r=1 n=3 a1=2 a2=4 s=oct
```

## 範例輸出 2

```
This number has been deleted.
This is a transitive verb.
This is a bad number.
```

## 測資說明

對於 10% 的測試資料， $n \leq 25$ ， $s = \text{dec}$ 。

對於 30% 的測試資料， $n \leq 25000$ ， $s = \text{dec}$ 。

對於 50% 的測試資料， $n \leq 2500000$ ， $s = \text{dec}$ 。

對於 100% 的測試資料， $n \leq 2500000$ ， $s$  不限制。

## 第 4 題：安東尼統計

執行時限：3 秒 配分：10 分

挺挺跟小可魚學壞後，每天過著跟小可魚一樣的日子，早上 8:00 準時到校，跟教官哈拉一下後，直接走向 302 電腦教室，準備他今天的無窮迴圈 while(1){程式→睡覺→DOTA}。

挺挺今天決定到「場廣食美」買它的午餐。他今天要觀察他的同學們總共買了幾樣不同的午餐、以及數量，還有他要預測商家可以賺多少錢。

但是「中一南」的學生們很奇怪，會先統計完每一樣有多少人要買後，一起跟店家說，挺挺則是因為和老闆娘熟識，可以後來再要求老闆娘幫他做，然後趁機和老闆娘聊天。

挺挺自己會在最後，選擇最多人買的商品，買一份自己吃，而他也會把自己納入計算。商家賺錢的算法很特殊，他賺的錢是每一個商品的[LEN\* (11-LEN)\* 數量]的總和，式子中的 LEN 代表該項商品的名稱長度，也就是字元數。

挺挺有一個很奇怪的偏好，就是他喜歡「後面」，(迷之聲：有其他意思嗎？歪頭～)也就是說如果有同樣多的情形，不管是挺挺要購買或是要輸出，都選擇後面出現的。

挺挺今天要你計算賣出最多和最少的商品名稱和數量，以及商家賺的錢。

### 輸入

每個測試檔案只含一筆測資。

第一行會給一個數字 n 代表商家賣幾種商品( $0 < n < 100$ )，

接下來 n 行代表商品名稱，和商家賣出的數量 k，

可以確定每一個名稱都是英文，且不超過 10 個字元，

數量 k 必大於 0，小於  $10^8$ ，兩者之間以一個空白隔開。

除了以上的資料外，不會有其他任何輸入。

### 輸出

印出賣出最多和最少的商品名稱和數量，並計算商家賺了多少錢。

前十個字元為商品名稱，靠左對齊，之後緊接著商品數量，

請參考範例輸出的格式和順序，輸出必須完全按照格式，

注意名稱要對齊(10 個字元)，再直接接著印出數量。

## 範例輸入

```
3
Anthonyyes 10
Lantwtw 120
Void 350
```

## 範例輸出

```
Void      351
Anthonyyes10
13288
```

## 第 5 題：社長讓我戳

執行時限：10 秒 配分：10 分

話說，在 2010 年 6 月 10 號，TFcis12 開始受到 SB 社長的統治。這一個外表看似高中生，內心卻是糟糕不已的社長，總喜歡看到別人就戳~~

於是他在一年一度的 TFcis 聖誕午會中，決定玩一個遊戲。

事情是這樣的，他要全部的 N 個小學弟排成一個圓圈，並且為每一個學弟編號(1、2、3.....N)。接著他用他寫出的亂數產生器產生一個數 M。他要 1 號同學對前方第 1 格同學(2 號)戳一下，接著 3 號同學對前方第 2 格同學(5 號)戳一下，接著 6 號同學對前方第 3 格同學(9 號)戳一下... 以此類推，每一次被戳到的同學下一個人要再向前方第 X 個同學戳一下(可能是自己戳自己)... 直到  $X=M$  為止。最後，被戳到最多次的人就是冠軍，若有被戳到相同次數，以編號少者為冠軍，可以戳糟糕的社長 2147483647 次~

### 輸入

第一行有一個數 T，代表有幾組測資。

接下來有 T 行每一行有兩個數 N、M，代表有幾個同學和 X 的最大值。

( $1 \leq T \leq 10$ ， $1 \leq N$ ， $M < 100000$ )

### 輸出

第一行輸出冠軍的同學編號。

第二行輸出冠軍的同學被戳到的次數。

### 範例輸入

```
2
3 3
4 3
```

### 範例輸出

```
2
2
1
2
```

### 提示

在第一個測資中， $N=3$ ， $M=3$ ，1 號戳 2 號，3 號戳 2 號，3 號戳 3 號(可以戳自己)。

所以答案為二號最多，被戳兩次。

## 第 6 題：更多數列

執行時限：10 秒 配分：20 分

因為 sb283 用程式產生了許多他喜歡的數列，而且 charlie 之前幫 sb283 寫了幾個程式，因此他決定產生幾個有趣的數列。

charlie(你也可以稱呼他為 c1h2a3r4l5i6e ) 好不容易用了一個複雜的公式，產生了一個特別的遞增數列( 這可花了他 6 天的時間，可見這幾天他電腦都沒關機 )。但是很不幸的，因為學校電腦中毒，而且在學校用完隨身碟後沒有掃毒，導致家中電腦中毒，而有部分資料遺失。

好不容易把電腦還原了，又從被刪除的資料把這個數列檔給救了回來，卻發現原先的檔案被分割成了許多部分。很恰好的，數列都還是維持著遞增的狀態，但急於把資料救回的 charlie 却發現原先電腦上的編譯器不見了，因此無法寫程式來組合數列，一時間也無法連上網路下載，於是想請你幫忙。

為了簡化問題，你只需要**把兩個數列合併**即可。同時，為了快速知道特定數字在第幾項，因此必須加入**搜尋**的功能。

### 輸入

第 1 行有兩個數字  $a$ 、 $b$ ，代表第一個數列共有  $a$  項，第二個數列有  $b$  項。

第 2 行是第一個數列，共有  $a$  項，每個數字間有一個空白。

第 3 行是第二個數列，共有  $b$  項，每個數字間有一個空白。

第 4 行有個數字  $n$  ( 最大 100000 )，代表接下來有幾個項目要搜尋。

之後會有  $n$  行，每行有一個數字，代表要搜尋的項目。

所有數字都可用 32-bit signed integer 儲存。

### 輸出

第 1 行請先輸出合併後的數列 ( 長度是  $a+b$  )，數字間請以空白分隔。

接下來對於每個搜尋的要求，輸出該數字在已合併的數列中的位置 ( 項目編號從 1 開始 )，如果找不到請輸出「Number Not Found」，若有多個項目，請由小到大輸出所有符合的項目。

行尾都不可有空白，輸出時請注意。

### 範例輸入 1

```
3 4  
1 5 13  
4 7 8 9  
3  
5  
7  
10
```

### 範例輸出 1

```
1 4 5 7 8 9 13  
3  
4  
Number Not Found
```

### 範例輸入 2

```
5 7  
2 7 35 67 67  
3 8 35 35 35 89 100  
5  
2  
35  
67  
100  
3
```

### 範例輸出 2

```
2 3 7 8 35 35 35 35 67 67 89 100  
1  
5 6 7 8  
9 10  
12  
2
```

### 測資說明

對於 40% 的測試資料， $a$ 、 $b$  在 1000 以內。

對於 100% 的測試資料， $a$ 、 $b$  在 100000 以內。

## 第 7 題：客廳的電燈

執行時限：3 秒 配分：15 分

話說這次題目充滿了心機與陷阱，心地善良的王彥決定跳出來解救大家，條件是請你也幫他解決一項難題，故事是這樣的：

班際排球賽快接近了，為了幫班上盡一份心力，王彥全心投入班上的練習，除了放學留下來打球，他在家裡也努力的練習托球，但是有一天悲劇發生了，他在練習舉球時不小心打到家裡客廳的電燈了，看著屋內唯一的光源在半夜閃爍，王彥下的全身發抖，過了不久，電燈終於停止掙扎，暗了下來。

拿了梯子上去檢查之後，王彥才發現他打破的是特殊定製的燈泡，在市面上是買不到的，想了很久之後他想起來在家裡的倉庫裡有去年聖誕節留下的 LED 燈飾，王彥為了不讓爸媽生氣，決定用 LED 燈組合起來偽裝成一般燈泡，以節省成本。

王彥苦思了很久，卻不知如何下手，只好向偉大的你求助了，希望你可以在他父母回來之前組好所有受損的燈泡，讓王彥度過這次難關吧。

已知每一個燈泡的光源皆可視為由各個不同波長之光波組成，彼此之間不會互相干擾，任一波長之光波強度有加成性，即不考慮光波之干涉、繞射效應，又因為 LED 燈發射之光波涵蓋範圍非常的小，所以假設每一個 LED 燈皆為單頻光，又因為王彥的父母眼力十分的好，所以組合出來頻譜需要和原本的燈泡光譜一模一樣，即任一波長對應的強度值應該要和各個 LED 在此波長的強度值的總和相等。

王彥發現每顆 LED 發出的光波強度均相同(訂為 1)，但是每個 LED 的耗電功率卻不同，為了避免電費暴增被發現，請你找出最省電的組合，且每個 LED 燈只能使用一次。

因為王彥想要自己動手組裝，所以你只要算出所有燈泡的最小耗電功率總和就可以了，你可以假設不會有無解的情形。

## 輸入

每個測試檔裡只有一筆測試資料，第一部分是破損燈泡的光譜資料，第一行有一個正整數  $N$ ，代表有  $N$  個破損的燈泡，接下來會有每個燈泡的光譜資料，彼此以一個空白行區隔，每個燈泡的光譜資料一開始會有一個正整數  $M$  ( $1 \leq M \leq 600$ )，代表有  $M$  個波長值，接下來  $M$  行每行有  $a_i$  和  $b_i$ ，代表波長  $a_i$  對應的強度值為  $b_i$ ，每個燈泡不會有重複的  $a_i$ 。

第二部分是 LED 的性能資料，第二部份第一行有一個正整數  $X$ ，代表有  $X$  個可用的 LED 燈，之後有  $X$  行，每行有兩個正整數  $p_k$ 、 $q_k$ ，代表第  $k$  個 LED 燈的發光波長  $p_k$ ，耗電功率  $q_k$ ，第一、二部分間有一個空白行做區隔。

## 輸出

輸出一個數代表所有燈泡的最小總耗電功率。

## 範例輸入

```
2
1
200 1

2
200 1
300 2

5
300 4
300 8
200 2
200 1
400 16
```

## 範例輸出

```
15
```

## 測資說明

對於 20% 的測資， $X \leq 15$ 。

對於 40% 的測資， $X \leq 5000$ 。

對於 60% 的測資， $q_k \leq 200000$ ，且答案可以用 32-bit unsigned int 儲存。

對於 100% 的測資， $q_k \leq 2147483647$ 。

$N, M \leq 600, X \leq 400000, 200 \leq a_i < 800$

## 第 8 題 ( 加分題 )：畢業旅行

執行時限：3 秒 配分：10 分

時光飛逝，學長們不知不覺在一中混過了一整年，升上了二年級，最期待的當然是畢業旅行拉~~正當大家興高采烈地討論時，學校原本安排的遊覽車公司居然惡性倒閉了，偉大的婷婷為了不讓大家的期待落空，決定扛下這個重要的任務，但是身為數學小老師的婷婷卻腦袋不太清楚，所以請你幫他算一下總共需要請幾輛車吧。

已知要參加的人數有 600 人，但不同車型能容納的人數卻不相同，請對每一個給定的容納人數輸出需要幾輛車才載的完。

### 輸入

輸入第一行有個正整數  $T$  ( $T \leq 10$ )，代表有幾個詢問。

接下來有  $T$  行，每行有個  $M$ ，代表一輛車可以載  $M$  個人，且  $M$  必為 600 的因數。

### 輸出

每行輸出需要幾輛車才載的完。

### 範例輸入

```
3  
2  
15  
6
```

### 範例輸出

```
300  
40  
100
```

### 提示

本題程式碼請命名為「PRO8」。