

# Software CRC

---

ShelterZone Code Contest S1 PracticeContest Q8

題目出處：[UVa 128](#)

題目翻譯：[Lucky貓的UVA \(ACM\) 園地](#)

## 題目

你是一家有很多個人電腦的公司上班。你的老闆，連一分都省博士，想要把這些個人電腦連線，但是他又不想要花錢買網路卡。由於你不小心告訴老闆每台電腦出廠時就有一個非同步串列埠在上面，老闆當然把腦筋動到這不用花錢的解決方案上。於是可憐的你被指派來完成這工作軟體的需求，以使這些電腦之間可以連線。

你已經讀了許多關於通訊的書並且知道在傳送及接收訊息時，確保訊息的正確性是一個重點。典型的作法是在訊息的最後加上額外的錯誤檢查碼。這個資訊允許接收程式檢查出傳送的資訊是否有錯誤（在大多數的例子）。於是，你跑到圖書館借回了一本關於通訊厚厚的書，利用週末（當然沒有加班費）研究錯誤檢查的部分。

最後，你決定CRC（cyclic redundancy check）是最適合的錯誤檢查方式。以下描述這個機制：

### CRC Generation

待傳遞的訊息被視為一系列長長二元數。訊息的第一個位元組（byte）被當作這二元數最重要的位元組，第二個位元組被當作第二重要的位元組，依此類推。這個二元數被稱為" $m$ "。當傳送時會在" $m$ "之後加上2個位元組的CRC檢查碼，然後整個二元數稱為" $m2$ "。

這個CRC的檢查碼被產生使得" $m2$ "可以整除某個16位元的值 " $g$ "。所以當接收端收到一訊息，只要把他除以 " $g$ "，如果餘數為0即代表此訊息正確。

你也注意到，大部分建議採用的 $g$ 值都是奇數，所以你決定用34943當作 $g$ 的值。現在你迫切的任務就是對待傳送的訊息，寫一個程式算出這個CRC的值。

例如：若要傳送的訊息為：AB（二進位的表示式為：01000001 01000010），你必須算出的CRC值應為601B（01100000 00011011的十六進位表示式），使得01000001 01000010 01100000 00011011可以整除34943（10進位）。

---

## 輸入

每組測試資料一行，含有一個待傳送的訊息（不會超過1024個ASCII字元的長度），列結束字元（End of line）不被視為待傳送訊息的一部份。若遇到只含#的一行，代表輸入結束（此列無須輸出）。

## 輸出

對每組測試資料輸出一行，CRC的值（以十六進位表示）。請注意：CRC的值一定介於0到34942（十進位）之間。

---

## 範例輸入

```
this is a test
```

```
A
```

```
AB
```

```
Nothing gonna change my love for you.
```

```
#
```

## 範例輸出

```
77 FD
```

```
00 00
```

```
0C 86
```

```
60 1B
```

```
57 98
```