April Fools Day Contest 2019

這是第一次 TIOJ 上有類似的競賽，感謝大家的參與（居然有將近兩千個 submission！），不知道大家玩得還開心嗎？當然從結果看來，大家可能還沒有參透這種風格，所以有很多梗都沒有人發現，這裡就來公布這次題目的解答吧！

前言

Contest description 裡面明明就有特別寫到要善用提問系統（事實上很多題都可以從提問的回覆得到明顯的提示），還在公告中提醒了兩次，為什麼還是沒有什麼人問問題呢 OAO

另外感覺 scoreboard 有一點把風向帶歪了 (?

pA. 找規律挑戰題

這是一類古老的題目，給出數列的前幾項，要你推斷出下一項是什麼。既然如此，打開 OEIS，搜尋 1,9,83，就出現了八個合理的數列，每一個的第四項都不太一樣，那就都試試看吧？

你有仔細看題目嗎？一般的題目如果說「你能寫個程式解決這個問題嗎？」，意思就是「請你寫個程式解決這個問題」，所以題目問說「你能不查 OEIS 就知道答案嗎？」，意思就是「請你不要查 OEIS 就寫這題。」可是 special judge 又怎麼會知道你有沒有查 OEIS 呢？最簡單的方法自然是看你輸出的數列有沒有出現在 OEIS 囉！因此你要做的事就是輸出一個不在 OEIS 中的數列。而任何第四項都是合理的，畢竟你總是可以找到一個三次多項式嘛！

pB. 半個岳飛

這是一題沒有題目敘述的題目，但是標題看起來很玄。什麼叫做「半個岳飛」呢？如果你對歷史夠熟，你會知道岳飛最著名的事蹟是他拿了十二面金牌 (?)，所以半個岳飛應該就是六面金牌。這個線索因此指向了在 IOI 拿過六面金牌的人：Gennady Korotkevich，或稱 tourist。

接下來可以發現輸入範圍也怪神奇的：$N \leq 147$。這個提示比較難發現，其答案是在 Codeforces 上：這場比賽開始前 tourist 在 Codeforces 上參與的 rated contest 數量剛好是 147，而範測給的那幾筆恰好都是他在那一場的 rating delta。至此題目就很明顯了：輸出 tourist 參與的某場 rated contest 的 rating delta 是多少。（手打 147 個數字有點累，所以把 CF 網頁的 source code 抓出來用程式整理也許是個比較好的選擇。）

pC. JEE

這題的題目非常簡單明瞭，就是單純的 boolean satisfiability problem，沒有任何懸念。

但等等，120 個變數的 SAT 到底要怎麼解呢？網路上有不少 SAT solver，所以抓一個來用或許是一個選擇。不過這裡有個關鍵：這個問題的標題叫做「JEE」。如果把 JEE 拿去 google 的話，你會發現 JEE 是印度的大學入學考試之一，也就是說這題實際上是印度版的 SAT！（這裡用到的是「SAT」同時代表美國的大學考試和布林可滿足性問題的雙關。而所謂「印度」的說法，是來自於一個傳說：傳說中印度的某些程式競賽，測資範圍都是玄學等級的：例如題目標明 $N \leq 10^6$，實際的測資可能是 $N \leq 10$。對於不知道這個傳說的人，至少你還是可以用正常的做法在這題拿到 62 分的 subtask，然後還順便多知道一個傳說，不錯吧！）

可是如果 assert 測資，你還是會發現 N 確實有到 120。但是這題的「印度」點不太一樣：事實上，出現在式子中的變數最多只有 21 個！因此，把那些變數找出來，然後爆搜做完就可以了。當然，這題中 and、or、xor、not、nand、nor、xnor 這七種常見的邏輯閘都有出現，如果你只考慮到 and、or、not 三種的話，你只會拿到 31 或 48 分。

當然，不考一點程式的小技巧是不行的。這題的爆搜其實有更有效率的做法：把每個變數壓成一個大小為 2^N 的 bitset，這樣就可以直接對這些變數做運算，然後看最後的結果哪一位是 0。不過這樣空間會爆炸，所以在處理運算的時候，要先往運算樹比較深的那邊走，且一個變數在和其他變數做運算的時候才把 bitset 產生出來，這樣可以確保只會用到 $O(\log K)$ 個 bitset。如果你沒有壓這兩樣常數的話，你只會拿到 96 分。

最後呢，感謝你耐心地看完上面那些題解！這題真正關鍵的地方在於輸出格式：為什麼題目要求每行輸出一個「字串」，明明範例輸出每行就只有一個字元？一個數位電路顯然可以在不同的時間有不同的輸入，所以這題是允許每行輸出一個 0/1 字串的（當然長度必須相同）！因此只要對於所有出現在電路中的變數，輸出所有可能組合，而對沒出現在電路中的變數輸出長度相同的 0/1 字串就可以了！

（這題的出題時間大概佔了準備整場比賽的 70% 吧……）

pD. 理春議 3

這題就是真正考驗生活常識與找規律能力的題目了！標題「理春議」只是代表輸入的數字是以十六進位的形式給的。

這題的解題關鍵是看懂範例測資。首先，你會發現輸入的測資有一點有趣：這些十六進位數只由 0、1、4、5、7、c、d 七種位構成。而且兩個字的就有一個 0、三個字的就有兩個 0。這並不是巧合：假如把輸入的十六進位轉換成二進位，會發現所有連續 0 長度只有 1、3、7 三種，而連續 1 的長度只有 1、3 兩種。再稍微數一下長度為 3、7 的連續 0，會發現它們的數量看起來很像是字母和字的分界。這已經構成了明顯的線索：事實上，題目就是輸入以十六進位表達的摩斯電碼（1 代表有訊號，0 代表無訊號，所以長度為 1、3 的連續 1 分別是 dot 和 dash），輸出解碼後的字串。

這題的提問系統的回覆也會附帶其對應的 input，可以做為額外的範測使用。可惜沒有人在這題提問。

（我原本以為這題會有不少人做出來的……）

pE. 生命、宇宙以及任何事情的終極答案

標題直接指向一個著名的數字 42。因此，你嘗試 `puts("42")`，然後獲得了 WA 42？

這題是本場中最難發現梗的一題（但是有很多人都花了一堆時間在傳這題，應該是受到部分分的影響）。首先，明明題目只有要求你輸出一個數字，但卻為甚麼只有得到一個 subtask 的分數呢？這時你必須相信題目，也就是答案是對的，而你仍然需要呼叫那兩個函數，但也許輸出的 code 或方式不太對。接下來，從出題時 TIOJ 的設計可以知道，一個題目若是 interactive，就無法採用 special judge，也就是說 judge 無法判斷任何與 code 相關的資訊，因此絕對是輸出方式的問題。

因此，關鍵點在於，什麼是「輸出的方式」呢？也許會想到輸出格式的問題，不幸的是這是個錯誤的方向。正確的方向是考慮不同的輸出位置：一個程式又不一定要輸出到標準輸出，還可以輸出到檔案、輸出到螢幕等等各種奇怪的地方啊！至於要輸出到哪裡才能得到 AC 呢？線索在於輸出前後都要呼叫 interactive 函數。輸出前需要呼叫 interactive 函數，代表那個要輸出的地方是需要執行一段程式才能開啟的，因此排除了直接輸出到檔案的可能性。剩下可以稱為「輸出」的地方，大概就只剩下 file descriptor 了吧！秉持著這個想法，稍微 assert 一下可以發現只有 1、2、42 是開著的，而最合理的顯然是 42。因此，把 42 輸出到 42 號 file descriptor，就能獲得 AC 了！

pF. Is this alphabet handstanding?

又是一題沒有題敘的題目。從標題和輸入、輸出格式看來，這個題目是要判斷每個字母是不是「handstanding」，也就是字母是不是「倒立」的。（特別用英文是為了強調這個「倒立」是真正意義上的倒立。）

那究竟什麼叫做倒立的字母呢？從範例測資可以知道其中三個字母是或不是倒立的，並且有八個字母保證不會出現在輸入中，然而仍然有十五個字母不確定。保證不會出現在字串中的八個字母看起來也沒有甚麼規律，只能看出大部分是英文中比較少出現的字。這題的關鍵是要想到這或許是某種密碼，然後到維基百科的「替換式密碼」（也就是逐字母替換的密碼）去查一下，就會在那篇的最底下發現一個很神奇的圖案，是描述一種把字母替換成火柴人的編碼「dancing men cipher」，而這個編碼中 d、g、t 三個字母編碼後的火柴人是倒立的。這也解釋了為什麼會有八個字母不會出現在輸入中，因為在原著《福爾摩斯歸來記．小舞人探案》中沒有提到這八個字母對應的符號。

當然，要找到這題的梗還有一個更簡單的方法：「盡量地使用提問系統，以幫助你更理解題意」（摘錄自 contest description）。在比賽中如果針對這題提問的話，只要格式合法，不管你問的是什麼問題，都會得到以 dancing men cipher 編碼的回應喔！（恰好三種回應都沒有包含不合法的字母。）

有些人沒有用提示就直接做出來了，我猜應該是有看過原著吧 (?

pG. 1KiB 的題目

這題比較不算是梗題，而是做法很奇怪的題目。首先第一個梗是要輸出的「這題題目敘述」指的並非 CF 的那題，而是 TIOJ 上的這題，因此要連同後面那段中文一起輸出（這也可以從題目特別要求用 UTF-8 輸出看出來，否則只有英文的內容沒有什麼編碼問題）。當你輸出了正確的題敘，你會得到 24 分以作為獎賞 (?

剩下的提示是標題「1KiB 的題目」，這代表的意義是你的 code 長度必須恰好是 1 KiB。但是等等，光題目敘述本身就超過 1 KiB 了？

要寫出這題，你必須記得這個比賽可以使用 TIOJ 上所有的程式語言。由於 Python 有內建一些壓縮、解壓縮的 library（如 lzma），因此可以把題敘先壓縮起來（具體來說要先轉換成 big-5 編碼，再用 lzma 的 `FORMAT_RAW` 壓縮）、轉換成 base85，然後放進 code 裡面。解壓縮的時候則需要把一些文字常數如 `FORMAT_RAW`、`FILTER_LZMA2` 換成數字。這樣的 code 長度至少可以壓到 1011 bytes，再加上一些註解補到 1 KiB 理論上就可以 AC 了。然而因為 TIOJ 的 bytes 計算目前有一點小問題（特別是換行的處理），所以可能要多嘗試加或刪幾個 byte 才會過。

pH. 零的安全感

又有一題題目敘述稍微正常一點的題目了！不過等等，函數 f 到底是什麼啊？

這題有個特殊的地方是它的分段給分。這也代表你可以隨便輸出幾個字串，然後看看所得到的分數是多少，據此推斷 f 到底是什麼函數（一開始題目敘述不小心把 8-bit 打成 8-byte 了 QQ）。事實上題目本身也有不少線索：首先是題目使用了「安全的函數」這個詞，還有 f 的值域是長度為 64 的字串。最後的結論是函數 f 是 SHA-256 的十六進位表達。

然而看一下 AC 這道題目的要求：$Q \geq 485$。等等，這不是相當於要構造出一個使得 SHA-256 至少有將近連續九十個 bit 是 0 的輸入，而這件事理論上只有大約 2^{-90} 的機率？對，~~所以這題你就拿部分分就好了~~但是正如俗話所說，「一台電腦不夠，那就用兩台；如果兩台還是不夠，那就把全世界的電腦都拿來用」(X)。一台電腦不夠，你有想到其實有個神秘組織包含了來自全世界的電腦，而這些電腦都在日以繼夜地協助你做這題嗎？沒錯，這個神秘組織就叫做 Bitcoin。因此你要做的事就是把 Bitcoin 一些區塊的 hash 資料載下來，然後發現高度 515910 這個區塊的 SHA-256 有連續 22 個 0 再加上另外兩個連續零，這相當於 $Q = 488$，就能 AC 了！另外，不要輸出多餘的空白或換行，不然 hash 的結果會完全不同，而且還有可能因為這樣超過 50 個字元的限制喔！

（一個有趣的事實：這題可以說是世界上需要最多計算資源才能 AC 的題目喔！）

這題提問系統也有提示：出題者和你的通信管道也被汪汪控制了，所以所有你收到的回覆都會是套用過 f 的訊息。

pI. Concise BBrainFxxk

看這個標題和範例測資，顯然這題和 BrainFxxk 是脫不了關係的，不如就把範例測資中看起來像 BrainFxxk 的程式執行看看吧？

事實上，執行了之後應該不會跑出什麼有用的東西。不過如果仔細觀察一下執行之後記憶單元的狀況，例如第一筆範例測資（Hey）：

1	0	0	1	0	1	0	0	3	0	0	2	0	3	0	1	0	⋯

這究竟是什麼呢？再看看輸入「Hey」的 ASCII，「H」是 72、「e」是 101，轉換成二進位之後看起來和這個記憶體內容有一點點像？事實上，「BBrainFxxk」這個名稱的第一個「B」是「Bit」的縮寫，也就是說 `<` 和 `>` 兩個指令的移動單位是一個 bit 而非一個 byte，而其餘指令仍然是以 byte 為單位。至此題目就很明顯了：輸入一個字串，輸出能輸出這個字串的 BBrainFxxk 程式。

這樣子每一個字元大概最多會需要 22 個指令輸出，或許能拿到 56 分或 87 分。但是要怎麼 AC 呢？再看看題目，「Concise」……也許是輸出的程式要短一點？對！要獲得滿分，你輸出的程式不能超過 1 KiB！但是輸入卻有接近 10 KiB 那麼大？

這就是這題的梗了。題目說輸入一個字串，你有想過這個「輸入」同時也會是 BBrainFxxk 程式執行的時候的輸入嗎？因此事情非常簡單：試試看 BBrainFxxk 的 EOF 到底是怎麼表達的，然後寫一個把輸入全部輸出出來的程式就好了。結論是 BBrainFxxk 的 EOF 是讀一個 0 進來，因此只要輸出`,[.,]`這個程式就可以了。由於輸入有限定是可視字元，所以不會有什麼問題。

繞了一個大彎，如果有想到梗的話，其實這題根本不需要知道 BBrainFxxk 到底是什麼嘛，直接把它當 BrainFxxk 寫就會過了！這題提問系統也有提示：回答會附上一個可以被執行的 BBrainFxxk 的程式，其輸出也會是回答，可以當成額外的範例測資。然而這題也沒有人提問。

pJ. 停看聽

也許這是你第一次見過題目是個聲音檔？

總之，當你聽完這個很簡單的題目，你會發現輸入格式根本沒有講清楚，而你或許也會嘗試幾種常見的輸入格式，但都得到了 WA。這時標題也告訴要「聽」也要「看」，題目卻沒有除了聲音檔以外的東西，因此合理的方向是研究這個聲音檔。

題目裡有個神奇的常數：3185327，而這個數字恰好是這個聲音檔的採樣點數目。另外這個聲音檔的格式是 32-bit，也符合題目敘述的數字範圍。因此，這題就是輸入這個題目敘述的檔案，然後輸出每個採樣點的平方和。Python 有內建處理 wav 檔的 lib，所以應該不會太難寫；而時限的部分，在本機算好再直接輸出答案就可以了。

這題的提問系統也是有提示的：所有的回答都會以聲音檔表示，它們的長度都相同，且採樣點的平方和也都相同。不過看起來有戳到這題提問的人好像都沒有特別注意到回覆的聲音檔莫名的長，或者沒發現這其實是個提示。

一些標程

pC. `https://ideone.com/sd3JVF`

pD. `https://ideone.com/kFdKhe`

pE. `https://ideone.com/MrryO4`

pG. `https://ideone.com/jtFXSY`、`https://ideone.com/NuScL8`（壓縮的 code）

pJ. `https://ideone.com/dNh1gf`（本機上計算的 code）