

2015 網際網路程式設計全國大賽

高中組初賽

- 本次比賽共 7 題，含本封面共 22 頁。
- 全部題目的輸入都來自**標準輸入**。
輸入中可能包含多組輸入，依題目敘述分隔。
- 全部題目的輸出皆輸出到螢幕 (**標準輸出**)。
輸出和裁判的答案必須完全一致，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 比賽中上傳之程式碼，請依照以下規則命名：
 1. 若使用 C 做為比賽語言則命名為 `pa.c`, `pb.c`, 以此類推。
 2. 若使用 C++ 做為比賽語言則命名為 `pa.cpp`, `pb.cpp`, 以此類推。
- `cin` 輸入經測試發現速度遠慢於 `scanf` 輸入，
答題者若使用需自行承擔因輸入速度過慢導致 `Time Limit Exceeded` 的風險。
- 使用 `scanf` 或 `printf` 處理長整數 (`long long int`) 時，請使用 `%lld`。
- 每一題的執行時間限制，請參考 Kattis 上的題目敘述。

2015 網際網路程式設計全國大賽

解題程式輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int cases;
5     scanf("%d", &cases);
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         scanf("%lld %lld", &a, &b);
10        printf("%lld\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main()
3 {
4     int cases;
5     std::cin >> cases;
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

A. 校園偶像計劃 II

Problem ID: song

去年穗香學園的眾人在大家的幫忙之下，成功地贏得了「我愛偶像」這場全國矚目的校園偶像大賽。然而，時光荏苒，由眾人一同選出的最佳九人中，有幾位原先已經三年級的學生現在即將要畢業了。在經過了一陣不捨和糾結的討論之下，他們決定光榮地解散這個充滿回憶的九人組合——畢竟，就是要這九個人才是當初令人充滿各種回憶的組合啊！

在最後的最後，穗香學園偶像研究社的這最佳九人，決定創作一首代表所有成員的歌曲，並以這首歌來當作最後一場演出的主題，同時為這一年來的青春回憶畫下完美的句點。而這首紀念之歌最大的特色將在於：「把成員的名字融入於歌詞當中，而且每個人的名字皆至少須在這首歌中出現兩次。」

然而，要如何把名字巧妙地嵌入歌詞當中而又不顯得生硬，確實不是一件容易的事。光編曲與填詞就讓眾人傷透了腦筋，更何況還需要設計舞蹈和安排演出的細節。因此，你決定再次幫助他們，找出最好的填詞方案。

對於這首歌中的每一句歌詞，我們都已經知道最多可以嵌入幾個名字，以及每個成員的名字對該句子的「合適程度」。並且，為了增加填詞的彈性，一句歌詞裡可以出現同一個人的名字不止一次。現在，給定每個人的名字在每個句子中的合適程度，請你寫一支程式計算出可能的最高「合適程度總和」。（若同一個人於一句歌詞中出現 k 次，則其合適程度也加總 k 次計算。）

這將是穗香學園初代團體的最後一次演出了，相信你幫忙編排的歌詞，會造就一場完美的演出！

Mu-s-ic, START!!

Input

輸入的第一行恰有一個正整數 N ，代表歌曲的句子數。

接下來將有 N 行，其中的第 i 行將有恰十個值 $S_i, h_i, e_i, k_i, u_i, r_i, m_i, n_i, f_i, y_i$ ，以單一空白字元隔開，分別依序代表第 i 句歌詞可以容納的名字數，以及九個人的名字嵌入第 i 句歌詞中的「合適程度」。

- $2 \leq N \leq 50$
- S_i 為整數且 $0 \leq S_i \leq 9$ ，對於所有 $1 \leq i \leq N$ 。
- $h_i, e_i, k_i, u_i, r_i, m_i, n_i, f_i, y_i$ 皆保證為整數或者是單一字元 'x'。若值為 'x' 即代表所對應到的名字不能置入句子 i 之中。若值非 'x'，則保證 $0 < h_i, e_i, k_i, u_i, r_i, m_i, n_i, f_i, y_i \leq 9999$ ，對於所有 $1 \leq i \leq N$ 。
- 保證至少存在一種符合要求的填詞方案。

Output

請輸出一行，恰包含一個整數，代表可能的最高「合適程度」總和。

Sample Input 1

```

3
8 20 10 10 10 10 10 10 10 10
9 2 3 2 2 1 1 3 1 3
4 1 1 1 1 1 1 1 1 1

```

Sample Output 1

```

158

```

Sample Input 2

```

7
1 x x 10 x x x x x x
1 x x x 10 x x x x x
2 10 x x x x x x x x
3 10 x 10 10 x x x x x
4 7 7 7 7 7 7 7 7 7
4 3 3 3 3 3 3 3 3 3
9 5 5 5 5 5 5 5 5 5

```

Sample Output 2

```

155

```

Sample Input 3

```

4
4 28 12 10 22 15 17 x 16 20
5 12 15 18 10 x 16 15 18 10
6 32 20 24 x 15 12 10 18 19
4 16 18 12 20 22 20 15 17 x

```

Sample Output 3

```

413

```

This page is intentionally left blank.

B. 盜墓迷城

Problem ID: treasure

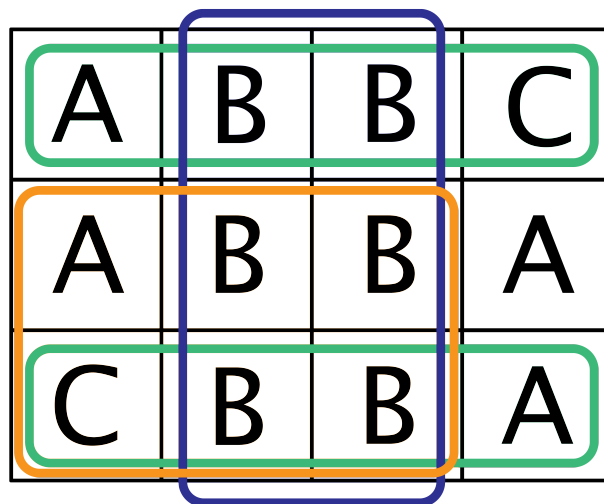
數百年前有一位盜墓專家艾迪，闖遍了各地大大小小的陵墓。

其實，早在秦始皇陵墓被挖掘出土前的幾百年，他就已經早先一步得知了陵墓的位置並闖了進去。陵墓內部暗道重重，地道峰迴路轉，暗房機關四佈。

艾迪在迷宮裡打轉了許久，好不容易在牆上發現了一處暗門。一推開，裡頭竟擺滿了大大小小的兵馬俑。仔細一看，它們排成了 N 列 M 行的矩形，其中大致上可分為步兵俑、弩兵俑以及騎兵俑。身為收藏狂的艾迪當然希望能把面前的兵馬俑都帶回去收藏，但此時他想起了他的祖傳盜墓守則，記載著以下的事情：**取走的兵馬俑原先一定要排成一個矩形，而且其中的步兵俑和騎兵俑數量要一樣，若違反了這條規則，整座陵墓就會因此陷落。**

雖然艾迪很想要收藏這些文物，但如果陵墓坍塌了他就會葬身於此。所以，艾迪想請你告訴他，在能安全脫身的情況下他最多能帶走幾座兵馬俑。

舉例來說（第二筆範例），下圖是一個 $N = 3, M = 4$ 的兵馬俑陣，其中 A 代表步兵， B 代表弩兵， C 代表騎兵。而圖中綠色矩形是一種可行的取法，可以帶走 4 尊兵馬俑，但不是最多的。圖中藍色或是黃色的取法皆可以帶走 6 尊兵馬俑，也是所有可行方法中最多的。因此艾迪最多能帶走 6 座兵馬俑。



Input

輸入的第一行包含兩個正整數 N, M ，以單一空白字元隔開，代表兵馬俑排列成 N 列 M 行。

接下來共有 N 行，其中每一行有一個長度 M 的字串 S_i ，表示第 i 列中的兵馬俑依序是哪一種類型，其中 'A' 代表步兵，'B' 代表弩兵，'C' 代表騎兵。

- $1 \leq N, M \leq 500$
- S_i 只會是 'A'、'B' 或 'C'。

Output

請輸出一行，恰包含一個整數，表示艾迪最多能夠帶走的兵馬俑數量。

Sample Input 1

```
3 4
AACA
AAAA
AAAA
```

Sample Output 1

```
2
```

Sample Input 2

```
3 4
ABBC
ABBA
CBBA
```

Sample Output 2

```
6
```


C. 爬樓梯

Problem ID: stair

運動系女孩——小希，每天回家的時候她都不搭電梯，而是選擇爬樓梯，順腳節能減碳愛地球。但是，每次都單純地爬樓梯實在是太枯燥了，富有求知慾的小希於是想到了一個有趣的問題：

「如果每步最多可以向上跨 m 階的話，有幾種方式可以從第 0 階爬到第 n 階呢？」

舉例而言，如果 $n = 4, m = 2$ 的話，有 $\langle 1, 2, 3, 4 \rangle, \langle 1, 2, 4 \rangle, \langle 1, 3, 4 \rangle, \langle 2, 3, 4 \rangle, \langle 2, 4 \rangle$ 這五種不同的過程可以從第 0 階爬到第 n 階。

然而，小希數著數著發現，在一些情況下方法數實在太多種了，希望聰明的你可以幫幫她，計算出共有多少種不同的爬樓梯過程。

Input

輸入有一行兩個整數 n, m ，以單一空白字元隔開，分別代表總共有幾階跟每步至多跨幾階。

- $1 \leq n \leq 10^{18}$
- $1 \leq m \leq \min(n, 500)$

Output

請輸出一個整數於一行，代表共有幾種不同的可行爬樓梯過程。由於答案可能很大，請輸出答案除以 $1000000007 (10^9 + 7)$ 的餘數。

Sample Input 1

4 2

Sample Output 1

5

Sample Input 2

32 32

Sample Output 2

147483634

D. 頗汪吃水餃

Problem ID: subseq

頗汪最喜歡吃水餃了！

今天頗汪在下課後來到了抬灣大學後門名聲響亮的「大諛迪水餃」。

「大諛迪水餃」可不是普通的水餃店。老闆諛迪有著他獨特的經營理念，他說過一句名言：「要做出水餃很簡單，但要做出讓顧客驚艷的水餃才是最重要的。」所以在「大諛迪水餃」，舉凡常見的高麗菜水餃、韭菜水餃……甚至是稀有的龍蝦水餃、鳳梨水餃、哈密瓜水餃……只要你想得到的口味，可以說是應有盡有！而每一種口味也有其相對應的編號，例如：0 號是常見的高麗菜口味、1021 號是雞肉口味，而 514514 號則是極其稀有的芒果口味。

大食量的頗汪為了能享用到盡量多種稀有口味的水餃，一坐下來就點了很多很多的水餃。熱騰騰的水餃上桌後，頗汪心想，只是吃水餃好像有點枯燥、沒什麼意思，所以他把面前的 N 顆水餃一一編號，打算要從第一顆依序吃到第 N 顆。但是頗汪是個善變的人，他隨時都可能因為心情不好而把筷子上的這顆水餃丟掉（但因為頗汪是來吃水餃的，他絕對不會把全部 N 顆水餃都丟掉）。不過基於對水餃的熱愛，在吃的過程中頗汪會在紙上依序記錄下他所吃下的每一個水餃的口味，而在他享用完所有水餃後，紙上的記錄便會形成一個水餃序列 (dumpling sequence)。

在開始享用眼前的水餃大餐前，頗汪想請你幫他算算看他最後可能得到的水餃序列 (dumpling sequence) 有幾種？

例如：頗汪點了 4 顆水餃，從第一顆到第四顆分別是高麗菜口味、韭菜口味、韭菜口味以及高麗菜口味，依照各口味的編號表示則是 0, 1, 1, 0，如果頗汪在吃第二顆水餃前突然因為反胃而把第二顆水餃丟掉，最後便會得到 $\langle 0, 1, 0 \rangle$ 的水餃序列。

而在以上的情況下，頗汪最後可能得到的水餃序列 (dumpling sequence) 共有以下 10 種： $\langle 0 \rangle, \langle 1 \rangle, \langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1, 0 \rangle, \langle 0, 1, 1 \rangle, \langle 1, 1, 0 \rangle, \langle 0, 1, 1, 0 \rangle$ 。

Input

第一行有一個正整數 T ，代表測試資料筆數，接下來包含 T 筆測試資料。

對於每筆測試資料第一行，包含一個正整數 N ，代表頗汪點了多少顆水餃。下一行會有以單一空格隔開的 N 個非負整數 a_1, a_2, \dots, a_N ，其中 a_i 代表第 i 顆水餃的口味編號。

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^5$
- 對於所有 $1 \leq i \leq N$ ，滿足 $0 \leq a_i \leq 10^9$

Output

對於每筆測試資料，請輸出一行包含一個整數，代表最終可能出現的水餃序列 (dumpling sequence) 有幾種。

因為答案可能很大，請輸出答案除以 $1000000007 (10^9 + 7)$ 的餘數。

Sample Input 1

```
2
4
1 0 1 1
1
1000000000
```

Sample Output 1

```
9
1
```

Sample Input 2

```
3
8
7 1 2 2 5 1 3 9
8
5 1 4 5 0 2 1 6
8
10 21 1 0 2 1 102 1
```

Sample Output 2

```
183
235
207
```

E. 零

Problem ID: zero

「噹噹噹噹～」上課鐘聲無情地響起。

「又是數學課啊，可是排列組合什麼的我早就會了，要做什麼好呢……」小希心想。

「一個正整數的**階乘**是所有小於等於該數之正整數的乘積。自然數 n 的階乘寫作 $n!$ 。」

看著課本上關於階乘的介紹，無聊的小希嘗試將一些階乘寫下來： $1! = 1$, $2! = 1 \times 2 = 2$, $3! = 1 \times 2 \times 3 = 6$, $4! = 24$, $5! = 120$, ..., $9! = 362880$, $10! = 3628800$, ...。寫著寫著，細心的小希發現當 $n!$ 的 n 逐漸變大的時候， $n!$ 的結尾零的個數也會逐漸變多！

因為老師教的內容實在太簡單了，小希決定設計更有挑戰性一點的問題給自己：

「存不存在 n 使得 $n!$ 的結尾恰好有 m 個零呢？若有多種可能的 n ，那麼最小的 n 是多少呢？」

Input

輸入的第一行有一個整數 T ，代表共有幾筆測試資料。

每筆測試資料只有一行，恰包含一個整數 m ，代表 $n!$ 結尾零的個數。

- $1 \leq T \leq 100000$
- $1 \leq m \leq 10^9$

Output

對於每筆測試資料請輸出一個正整數 n 於一行，使得 $n!$ 的結尾恰好有 m 個零。如果有多種可能的話請輸出最小的 n 。

如果不存在 n 使得 $n!$ 的結尾恰好有 m 個零，請輸出 -1 。

Sample Input 1

4
1
2
5
10

Sample Output 1

5
10
-1
45

F. 秋刀魚捕撈季

Problem ID: fish

如果提到秋季的魚類，「秋刀魚」對於位在北太平洋附近的島國來說，是絕對重要的經濟魚類之一。

瑞鶴村是一個位於某個海島國家的沿海小村落，自古以來便以漁業維生。這個村莊裡的年輕人，如果畢業後不是移居外地求發展，就幾乎是會留在故鄉一起為大家的生計打拚。幸運的是，近年來由於捕魚科技的進步以及交通運輸的發展，眾人的生活也隨著現代化的腳步一步步地改善了許多。

曉涵是標準土生土長的瑞鶴女孩，從小就看著身為船長的父親出海的背影長大。最近由於秋刀魚季節的來臨，村莊內的各戶人家都忙得不可開交——畢竟，秋刀魚祭的收穫就佔了全村整年收入的大半。因此，今年曉涵決定待在家鄉一起幫忙。

整個秋刀魚祭為期數週，基本上這段期間內漁民們都會不斷出海捕撈秋刀魚。然而，為了永續經營，目前瑞鶴村對於捕撈秋刀魚訂立了一些規則：

- 基於「永續發展」的觀念，對於重量小於 M 公克的秋刀魚皆視為魚苗，一律不可捕撈。若撈到須放生回海中。
- 為了漁場資源的均分，每艘船每次出海最多只能選擇帶回 F 條秋刀魚，多的數量皆須放回海中。

在曉涵爸爸的捕魚筆記中，清楚地記錄了每一隻捕撈上來的魚之重量。曉涵看著父親的去年的捕魚筆記，突然覺得似乎哪裡怪怪的……在比對了父親最後帶回來的秋刀魚總重量之後，曉涵覺得當初應該有更好的選法，使得最後帶回的漁獲總重量更多。

現在，給定曉涵父親捕魚筆記中的每條記錄，請幫曉涵算出她父親**最多**應該可以帶回多少總重量的秋刀魚。

Input

輸入的第一行恰有一個正整數 N ，其中 N 代表捕魚筆記中總共記錄了幾條魚的資訊。第二行則恰有兩個正整數 M, F ，以單一空白字元隔開，定義請參閱題目敘述。

接下來將有 N 行，其中的第 i 行代表按照順序的第 i 條魚之資訊。對於每一行，其中可能恰有一個字串「others」（不包含引號）代表該條魚並非秋刀魚。否則，將恰有一個正整數 w_i 代表該條魚的重量（單位為公克）。

- $1 \leq N \leq 500$
- $1 \leq M \leq 300$
- $1 \leq F \leq 500$
- $1 \leq w_i \leq 300$ ，對於所有 $1 \leq i \leq N$ 。

Output

請輸出一行恰包含一個整數，即根據所給的漁獲資訊，曉涵她父親最多應該可以帶回多少總重量（單位為公克）的秋刀魚。

Sample Input 1

```
4
90 2
125
100
85
105
```

Sample Output 1

```
230
```


Sample Input 2

```
3
50 3
175
others
50
```

Sample Output 2

```
225
```

Sample Input 3

```
5
150 500
140
160
300
160
140
```

Sample Output 3

```
620
```

This page is intentionally left blank.

G. 技能施放問題

Problem ID: skill

你最近發現一款名為〈狗狗貓貓大冒險〉的遊戲。這是一款結合愛與勇氣而且擬真度極高的第一人稱角色扮演遊戲。在這款遊戲中，你所將扮演的角色是赤貓，因此你將能在遊戲中做到諸如：翻滾、跳躍、從高空降落、玩貓砂、跳火圈、捉老鼠、操縱人類等等，一般貓咪可以做到的事。

但在〈狗狗貓貓大冒險〉中，還加入了更多元素。的確，到目前為止都還沒提到狗狗的部分。事實上，這款遊戲將使你沉迷的最主要原因在於，這「狗狗」並不是名詞，而是形容詞！你所操縱的赤貓還擁有更多強大的技能，像是游泳、搖尾巴、導盲、發呆、搜查等，儼然是融合狗與貓的最強究極體。事實上，還有更多待解鎖的技能等你去發掘！於是你迫不及待地開始玩〈狗狗貓貓大冒險〉。

進入遊戲之後，你發現赤貓竟然只是一隻小貓咪。在遊戲中赤貓每秒能累積一點經驗值，隨著經驗值的累積，赤貓才會慢慢成長。接下來，你遇到的第一個任務是要「尋找食物」來填飽肚子，於是你馬上按下赤貓的技能「操縱人類」，希望能獲得食物。但螢幕上卻顯示：「不具備所需的能力值！」

這時你才發現，原來要施放技能，必須具備某些特定的能力值，而每項技能需要的能力值又不盡相同。此外，在遊戲中獲得能力值的唯一方法就是花費已累積的經驗值。由於赤貓有過人的記憶力，一旦獲得一項能力值便永遠不會忘記。

於是你心想，難道只能四處翻找食物，慢慢等經驗值累積嗎？這時聰明的你發現，在螢幕右下角有個小小的按鍵，上頭寫著「直接進入魔王關！」興奮的你毫不猶豫的按了下去！

進到魔王關，你眼前忽然冒出一隻看起來就是大魔王的神祕生物，手足無措的你突然發現一旁的所有技能都已經解鎖了！

而螢幕上跳出了一句提示：「只要施放出所有技能，便能打敗大魔王！」

情急之下，你以迅雷不及掩耳的速度……上上下下左左右右 ABCABC……把所有技能都按了一遍。但此時的赤貓根本還不具備任何能力值，而且已累積的經驗值也歸零了，理所當然是一項技能都沒施放成功。

由於著急沒什麼用，所以你只好強迫自己冷靜下來。然後你就發現，在魔王關中，如果想施放第 i 項技能，無論缺少哪些能力值，你都可以花費相對應的 C_i 點經驗值，一次獲得這些缺少的能力值。

大魔王正朝你步步近逼，你所剩的時間不多了。因此，你想找出一個施放技能的順序，讓你能夠在最短的時間內照著這個順序使出所有技能。

遊戲內以不同的大寫英文字母代表不同的能力值。舉例來說（即第一筆範例測試資料），你現在必須施放出編號為 1, 2, 3 的三項技能，而第一項技能需要的能力值為 AB、 $C_1 = 3$ ，第二項技能需要的能力值為 A、 $C_2 = 1$ ，第三項技能需要的能力值為 B、 $C_3 = 1$ ，則以下為不同施放順序所需要花費的經驗值：

1. $\langle 1,2,3 \rangle$ ：需花費 3 點經驗值。（施放技能一時花 C_1 獲得能力值 AB）
2. $\langle 1,3,2 \rangle$ ：需花費 3 點經驗值。（施放技能一時花 C_1 獲得能力值 AB）
3. $\langle 2,1,3 \rangle$ ：需花費 4 點經驗值。（施放技能二時花 C_2 獲得能力值 A，再施放技能一時再花費 C_1 獲得能力值 B）
4. $\langle 2,3,1 \rangle$ ：需花費 2 點經驗值。（施放技能二時花 C_2 獲得能力值 A，再施放技能三時再花費 C_3 獲得能力值 B）
5. $\langle 3,1,2 \rangle$ ：需花費 4 點經驗值。（施放技能三時花 C_3 獲得能力值 B，再施放技能一時再花費 C_1 獲得能力值 A）
6. $\langle 3,2,1 \rangle$ ：需花費 2 點經驗值。（施放技能三時花 C_3 獲得能力值 B，再施放技能二時再花費 C_2 獲得能力值 A）

第四以及第六種施放順序都只需要花費兩點經驗值，能在最短時間的收集完成，因此，兩種施放順序都為最佳技能施放順序。

Input

第一行有一個正整數 T ，代表接下來有 T 筆測試資料。

每筆測試資料第一行，包含兩個正整數 N, M ，代表總共有 N 項技能，且這 N 項技能所需的能力值種類不超過 M 種。接下來包含 N 行，每行包含一個正整數 C_i 以及一個字串 S_i ，代表施放第 i 種技能時，需要 S_i 中的各種能力值（每個字元即代表不同能力值），而且可以花費 C_i 點經驗值獲得 S_i 中你尚未獲得的每一項能力值。

- $1 \leq T \leq 25$
- $1 \leq N \leq 10^5$
- $1 \leq M \leq 12$
- $0 \leq C_i \leq 10^8$
- $1 \leq |S_i| \leq 12$ ，並且所有的字元皆不相同，只包含大寫英文字母。

Output

對於每筆測試資料，請輸出兩行。

第一行請輸出一個整數，代表最少需要多少時間才能使出所有技能。

第二行輸出一個 1 到 N 的排列（數字間以空格隔開），代表一組最佳技能施放順序。如果有多組技能施放順序能在最短時間內使出所有技能，請輸出**任意**一種技能施放順序。

Sample Input 1

```

2
3 2
3 AB
1 A
1 B
3 2
3 AB
1 A
1 B

```

Sample Output 1

```

2
2 3 1
2
3 2 1

```

Sample Input 2

```

2
2 2
1 A
2 Z
2 2
2 AB
1 A

```

Sample Output 2

```

3
1 2
2
1 2

```

Sample Input 3

```

3
1 4
2015 NPSC
3 10
1 RANK
7 AC
514 PENALTY
2 4
100000000 GL
100000000 HF

```

Sample Output 3

```

2015
1
522
2 1 3
200000000
2 1

```