

2011 網際網路程式設計全國大賽 國中組決賽

- 題目：本次比賽共七題（含本封面共 20 頁）。
- 題目輸入：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。
- 題目輸出：全部的輸出皆輸出到螢幕(**標準輸出**)。
- 時間限制：每一題的執行時間限制如下表所示。其間執行的電腦上不會有別的動作、也不會使用鍵盤或滑鼠。
- 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`, 以此類推未按照此規則命名之程式碼將可能因此得到 `Compilation Error`。
- `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。

表 1: 題目資訊

	題目名稱	執行時間限制
題目A	賓果遊戲	5 秒
題目B	分組競賽	10 秒
題目C	圈圈國	5 秒
題目D	簡單題的逆襲	10 秒
題目E	天王星計畫	10 秒
題目F	田忌賽馬外傳	5 秒
題目G	數據加密	10 秒

2011 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：

```
#include <stdio.h>
int main(void){
    int cases, i;
    double a, b;
    scanf("%d", &cases);
    for(i = 0;i < cases;i++){
        scanf("%lf %lf", &a, &b);
        printf("%.2f\n", a+b);
    }
    return 0;
}
```

C++ 程式範例：

```
#include <iomanip>
#include <iostream>
using namespace std;
int main(void){
    int cases, i;
    double a, b;
    cin >> cases;
    cout << fixed << setprecision(2);
    for(i = 0;i < cases;i++){
        cin >> a >> b;
        cout << a+b << endl;
    }
    return 0;
}
```

題目 A 賓果遊戲

執行時間限制: 5 秒

在整理櫃子的時候，創創和守守發現了一疊賓果卡（ 5×5 的方格紙上不按順序地印著 01 到 25 的數字），所以他們還沒整理完櫃子就開始玩起賓果遊戲來了。

03	21	05	15	08
14	10	17	23	04
18	06	22	12	09
24	20	13	07	16
11	25	01	19	02

22	12	15	03	16
11	23	20	14	19
02	17	09	05	07
24	04	10	21	01
06	13	18	25	08

04 14 07 22 02 01 09 17 05 18 19 12 08 21 15 06 10 16 03 24 20 25 13 23 11

創創和守守玩賓果遊戲的方式可能和我們有點不一樣：他們會一人先抽一張賓果卡，再使用從網路上找到的某個程式輸出一組數列（這個數列包含 25 個數字，每個數字都在 01 到 25 之間，已經出現過的數字就不會再重複出現），然後按照數列裡數字出現的順序把賓果卡上的數字劃掉，直到其中一人的賓果卡上出現一條長度為 5 的直線、橫線或斜線為止，達成連線的人就是贏家；如果兩個人同時達成連線，則視為平手。

仔細想想，每次都要尋找數字、再把它劃掉、還有判斷是否有人勝利，這一連串的動作其實冗長又重複，所以創創和守守現在只想知道是誰贏了遊戲，而且他們還要回去整理櫃子呢！你可以在他們整理完櫃子之前幫忙寫一個程式，判斷誰抽到的賓果卡會贏得遊戲嗎？

■ 輸入檔說明

第一行有一個整數 T ($T \leq 50$)，代表接下來有幾組測試資料。

每組測試資料包含兩張賓果卡和一組數列：前 10 列每列有 5 個數字，第 1 列到第 5 列代表第 1 張賓果卡，第 6 到第 10 列代表第 2 張賓果卡；第 11 列是由 25 個數字所組成的數列，每個數字都在 01 到 25 之間，出現過的數字不會重複出現。

■ 輸出檔說明

對於每組測試資料輸出哪張賓果卡會贏得遊戲，“1”代表第 1 張，“2”代表第 2 張，“0”代表平手。

■ 範例輸入

```
2
03 21 05 15 08
14 10 17 23 04
18 06 22 12 09
24 20 13 07 16
11 25 01 19 02
22 12 15 03 16
11 23 20 14 19
02 17 09 05 07
24 04 10 21 01
06 13 18 25 08
04 14 07 22 02 01 09 17 05 18 19 12 08 21 15 06 10 16 03
24 20 25 13 23 11
03 21 05 15 08
14 10 17 23 04
18 06 22 12 09
24 20 13 07 16
11 25 01 19 02
22 12 15 03 16
11 23 20 14 19
02 17 09 05 07
24 04 10 21 01
06 13 18 25 08
04 14 07 22 02 01 09 17 13 05 18 19 12 08 21 15 06 10 16
03 24 20 25 23 11
```

■ 範例輸出

2 0

本頁留白。

題目 B 分組競賽

執行時間限制: 10 秒

自古至今，卡恩國境內流傳著一種在當地十分盛行的遊戲——「番茄蹲」，而「番茄蹲」這個遊戲也慢慢地傳到了其他的國家，甚至被評選為上個世紀最流行的團康遊戲。該遊戲的規則如下：

每場「番茄蹲」遊戲中都有一個主持人，我們稱他為「卡恩」，而 N 位參與遊戲的人分別從 1 號開始編號到 N 號，每個參與者有「蹲下方」（蹲著）跟「蹲上方」（站著）兩種狀態，並且剛開始時全部的人都為「蹲上方」。主持人每個回合都會喊出：『卡恩蹲！卡恩蹲！卡恩蹲完換 k 蹲！』，這時編號為 k 的倍數的遊戲者就必須改變狀態（原為「蹲上方」的變成「蹲下方」，反之亦然）（ k 不會超過 N ）。對於每一回合，失誤（沒有依照「卡恩」的指示行動）的人便會被淘汰。而這遊戲將會進行 M 回合（ M 為某個規定好的正整數），結束後那些在這 M 回合中被淘汰的人們將要被懲罰（例如請存活下來的人喝飲料之類的）。

而現在，有 G 群人想要來一起玩「番茄蹲」，其中第 i 群人有 g_i 個人，而總共加起來有 N 個人（也就是說 $g_1 + g_2 + \dots + g_G = N$ ）。在玩之前，他們想要來訂一個新規則決定勝負：他們決定分成兩組人馬 A 隊和 B 隊，之後比積分等等（幸好這不是我們所在意的）。

此時，好奇的時嗣鳴同學產生了一個問題：他想知道這 G 群人究竟會有幾種可能的配置方案。但是，其中他只在意 **A 隊的人數減掉 B 隊的人數後除以 K 的餘數等於 R 的方案**。到底有幾種可能的方案呢？寫個程式解決這個問題吧！

需要注意的是，這 G 群人之中任何一群的所有人都一定要被分在同一組之中，**不可以被拆分**（也就是，對於同一群人，不能有的人被分到 A 隊有的人被分到 B 隊）。另外，我們定義兩個方案為「不同」的方案僅當存在某一組人在其中一種方案中被分到 A 隊而在另一種方案中被分到 B 隊。為了進一步簡化問題，我們也考慮其中一個隊伍為空的方案（意即我們可以把所有人都分到 A 隊或 B 隊——而這兩種為「不同」的方案）。

在本題中， a 除以 b 的餘數 r 定義為：存在一個整數 q ，符合 $a = bq + r$ ，其中 $0 \leq r < b$ 。例如 10 除以 7 的餘數為 3，而 -14 除以 9 的餘數為 4。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 50$)，代表接下來有幾組測試資料。

每一組測試資料的第一行有三個整數， G ($1 \leq G \leq 514$)、 K ($1 \leq K \leq 10000$) 和 R ($0 \leq R < K$)，意思分別如題目所述。

接下來第二行有共 G 個正整數 g_i ($1 \leq g_i \leq 10^6$)。

■ 輸出檔說明

對於每筆測試資料請輸出一個整數，代表所求的分組方案數。
由於答案可能很大，請只要輸出答案除以 1000000007 的餘數即可。

■ 範例輸入

```
3
3 5 0
1 2 2
2 8 1
2 3
2 7 2
2 3
```

■ 範例輸出

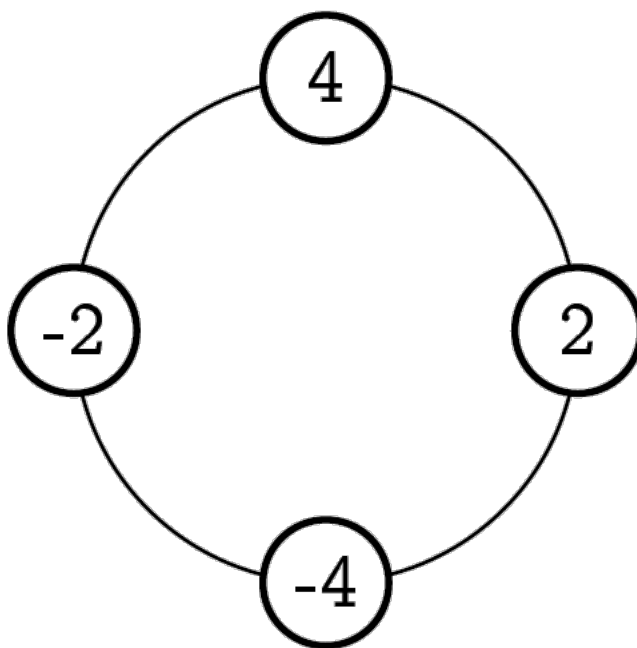
```
2
1
1
```


題目 C

圈圈國

執行時間限制: 5 秒

圈圈國的國土只有一個圈圈，也就是一個環。



在這個環上的每一格可以是一個民宅或是一個水井。因為每個水井的大小不同，所以水井可以供給的人數也不盡相同。而現在，圈圈國裡的水井不多不少剛好可以供給全國的國民使用。

可惜的是，圈圈國的圈圈王是個暴君，他不要讓人民有太多的接觸。因此，他希望在這個環上的格與格之間蓋城牆將人民隔開。但是，每個人在不越過城牆前提之下依然要有保有足夠的水井供給。圈圈王希望能夠利用這些城牆將人民隔成最多區塊。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 200$)，代表接下來有幾組測試資料。

每一組測試資料的第一行有一個整數 N ，代表圈圈國的國土上的格子數，而圈圈國的國土最少有兩格，最多一萬格。第二行有 N 個數字(A_1, A_2, \dots, A_N)，而且 $0 < |A_i| \leq 10000$ ，按照順序為環上每一格的民宅裡的人數或水井的大小。如果 $A_i > 0$ ，則代表這一格是民宅，而這個民宅裡的人數為 A_i 人。如果 $A_i < 0$ ，則代表這一格是水井，而這個水井可以供給 $|A_i|$ 人。

■ 輸出檔說明

對每筆測試資料輸出一個整數，代表圈圈王可以利用城牆所隔成的最多區塊數。

■ 範例輸入

```
4
4
4 2 -4 -2
5
1 2 -3 3 -3
5
2 -3 4 -4 1
2
1 -1
```

■ 範例輸出

```
1
2
2
1
```

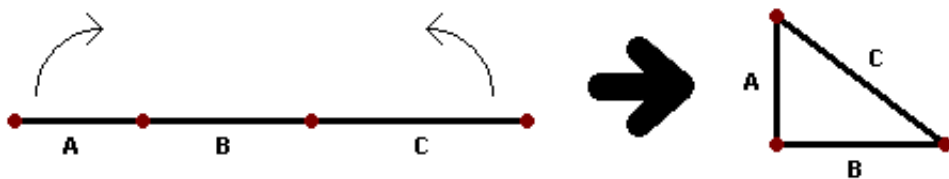
題目 D

簡單題的逆襲

執行時間限制: 10 秒

阿里不達王國一年一度的智者選拔舉辦在即，選拔方式是由王國頒布題目讓所有參加者作答。由於在選拔中能取得優秀成績的人將會受到王國的重視，所有國民都會參加，也因此題目的鑑別度顯得至關重要。

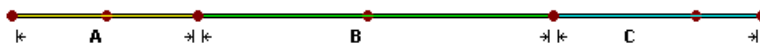
前任選拔的冠軍阿里依據傳統出任智者選拔的出題者，他需要同時準備簡單與困難的題目。阿里畫了一張圖如下並且想了兩個題目。第一題是給出 A, B, C 三條邊的長度算三角形的面積，第二題是只給出 $A + B + C$ 的長度算三角形的最大可能面積。



我們可以利用下列公式解出第一題， $area$ 便是答案，也可以從中推導出第二題的答案。

$$area = \sqrt{S \times (S - A) \times (S - B) \times (S - C)}, \quad S = \frac{A+B+C}{2}$$

在正式選拔開始之前，阿里找了一些朋友來測試解題，結果發現這份題目的鑑別度非常低。但他已沒時間重想題目，只能修改現有的題目。他決定把第二題給出的長度打散成 N 段，讓參加者選出左側任意連續段作為 A 、右側任意連續段作為 C 、剩餘部分作為 B ，下圖是 $N = 6$ 的一個範例。



倉促地把題目修改完，阿里還必須提供正確答案給王國以供比對，但他驚覺自己並不會修改後的題目！雖然他知道可以把所有選法都試過一遍，可是這不可能趕得上選拔時間，請你寫一個程式來幫助他快速地計算答案。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 100$)，代表接下來有幾組測試資料。

每組測試資料共有兩行，第一行包含一個整數 N ，下一行包含 N 個正整數表示各段的長度，各段長度不超過 100。 $(3 \leq N \leq 100000)$

■ 輸出檔說明

對每筆測試資料輸出一個浮點數，代表三角形的最大可能面積，答案需四捨五入至小數第二位。如果沒有任何選法能夠產生三角形請輸出 0.00。

■ 範例輸入

```
3
3
1 1 2
3
3 4 5
5
20 5 5 5 20
```

■ 範例輸出

```
0.00
6.00
139.05
```

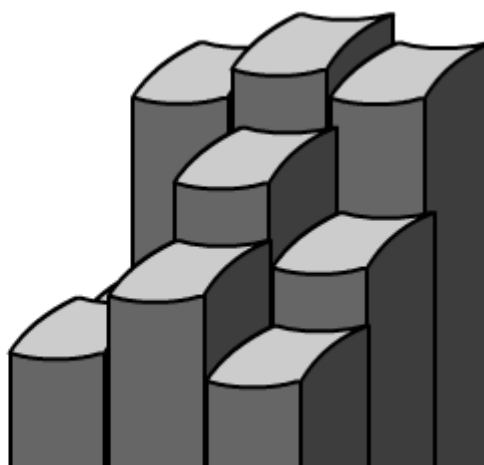
題目 E 天王星計畫

執行時間限制: 10 秒

在 2055 年，在地球的人類再次發射太空船，穆浩芙娜號，以及一顆衛星飛到天王星。這次的任務是去取得能源之石，當有了這塊寶石，對於地球上的人們，能源不再是問題，使用能源也更環保。在前一次的探測作業中，人們取得了能源之石的基本資訊，也就是在天王星的哪一處以及那附近的地理資訊。這次，穆浩芙娜將執行它的任務，去取得能源之石。

這次任務的內容，是先讓穆浩芙娜降落在天王星上，能源之石附近。爲了讓穆浩芙娜能瞭解地型，所以發射了伴隨它的一顆衛星來幫助穆浩芙娜。這顆衛星有兩個功用，其中一個就是能將穆浩芙娜附近的地形傳給它，以讓穆浩芙娜決定怎麼行走。能源之石附近的地形很特別，這邊皆是由很多的正方形隔成一塊一塊，但每個正方形的高度卻不一樣。穆浩芙娜它能行走以及爬升或下降 D 的高度，因爲它爬升和下降的距離是有限的，若兩個正方形高度相差太多，則穆浩芙娜就不能到達下一個正方形。下圖是一個例子，假設穆浩芙娜現在在中間那一格，也就是高度 7，且我們設 $D = 3$ ，那麼它可以往上、下、右移動，但它不能往左移動，因爲往左移動的高度差距爲 5。

8	9	8
2	7	4
3	5	2



所以，因為降落的地方不好控制，可能降落到一個不好的地方，又因為地形上的阻隔，穆浩芙娜有可能無法到達能源之石所在地。因此，這要出動衛星的第二個功能：升降光束！升降光束可以將一個完整的正方形區域變成電梯，我們再拿上面的圖當作例子，假設衛星照了一道升降光束在中間那一格，且假設穆浩芙娜在左邊那一格，則穆浩芙娜可以搭電梯：電梯先降到和 2 等高，然後升到 4，最後穆浩芙娜就可以到達右邊那一格。

可是呢，升降光束是很耗能源的！所以我們必須很節省地用。因此，計畫的工作團隊想請問你，我們現在有了地型高低的資訊，而至少要幾道光束才能成功的採取能源之石呢？請注意，你也能對穆浩芙娜降落的地點以及能源之石的所在地發射升降光束！

■ 輸入檔說明

測試資料第一個數字為 T ($T \leq 200$)，代表接下來會有幾個地圖。

對於每個地圖，第一行有兩個數字 M, N ，表示著地圖的大小，每行有 M 個數字，共 N 行 ($1 \leq M, N \leq 30$)。第二行有兩個數字 Sx, Sy ($1 \leq Sx \leq M, 1 \leq Sy \leq N$)，代表穆浩芙娜的開始座標。第三行有兩個數字 Px, Py ($1 \leq Px \leq M, 1 \leq Py \leq N$)，代表能源之石的座標。第四行有一個數字 D ，代表穆浩芙娜能往相鄰且高度差為 D 的正方形行走 ($0 \leq D \leq 100$)。最後的 N 行，每行有 M 個用空白隔開的整數，為每個正方形的高度，且介於 0 到 99 之間(包含 0 和 99)。

請注意，地圖的左上角座標為 $(1, 1)$ ，而右下角為 (M, N) 。

■ 輸出檔說明

對於每一個地圖，輸出一個整數表示需要幾個升降光束。

■ 範例輸入

```

3
6 4
6 3
1 2
4
1 1 1 6 6 6
1 1 6 6 6 1
1 6 6 6 1 1
6 6 6 1 1 1
3 3
1 2
2 3
10
1 2 3
2 3 4
3 4 5
15 20
3 2
11 15
7
19 14 48 48 12 34 35 35 11 37 11 21 45 32 37
46 41 14 32 43 22 32 31 46 48 25 17 16 21 28
39 42 37 29 45 24 33 21 12 23 43 34 31 41 23
38 37 14 32 47 47 29 13 31 19 28 46 45 40 12
18 36 10 32 34 18 16 15 20 19 20 40 16 31 43
38 39 33 14 44 46 10 16 26 21 38 34 49 36 13
27 28 48 12 19 31 43 45 49 28 34 20 27 36 43
36 31 35 14 42 40 39 27 23 27 42 36 20 11 46
45 17 25 24 41 22 40 40 31 14 16 40 37 41 17
27 27 17 43 33 35 39 19 48 31 38 32 36 36 40
23 18 30 41 12 25 20 49 34 27 38 13 45 11 12
40 21 46 24 30 46 31 38 49 38 14 11 24 23 49
28 28 30 38 17 37 18 23 18 13 17 31 40 27 23
44 39 26 25 21 10 19 49 46 48 33 34 18 14 19
39 32 45 15 23 33 13 37 44 33 18 10 28 28 30
46 48 11 39 48 19 47 42 32 28 22 48 29 40 27
48 41 25 18 46 41 12 44 42 25 18 16 12 16 45
43 39 32 14 26 11 48 12 21 31 29 47 46 22 39
45 42 10 40 13 39 31 18 11 39 25 33 12 35 32
15 18 36 27 47 42 22 19 44 11 39 26 19 18 22

```

■ 範例輸出

2
0
5

題目 F 田忌賽馬外傳

執行時間限制: 5 秒

齊使者如梁，孫臏以刑徒陰見，說齊使。齊使以爲奇，竊載與之齊。齊將田忌善而客待之。忌數與齊諸公子馳逐重射。孫子見其馬足不甚相遠，馬有上、中、下輩。於是孫子謂田忌曰：「君第重射，臣能令君勝。」田忌信然之，與王及諸公子逐射千金。及臨質，臏曰：「今以君之下駟與彼上駟，取君上駟與彼中駟，取君中駟與彼下駟。」既馳三輩畢，而田忌一不勝而再勝，卒得王千金。於是忌進孫子於威王。威王問兵法，遂以爲師。

—『史記。孫子吳起列傳第五』

千年以前，孫臏靠著過人的智謀，巧妙地調整比賽順序，讓三戰皆墨的田忌翻身成兩勝一敗的贏家，也爲自己贏得尊敬和重用。千年以後的今日，賽馬依然是熱門的活動，不過今天你要面對的是更困難的問題。

你和對手各有 N 匹馬，要進行 N 場比賽。一匹馬只限出場一次，同場比賽中速度較快的馬獲勝。若兩匹馬速度一樣，則算平手。你可以決定你的馬匹的出場順序；而你的對手，就如同齊王，會在第一場比賽出速度最快的馬，第二場出次快的馬， \dots ，第 N 場出速度最慢的馬。

除此之外，你還可以決定比賽的時間，全部 N 場比賽都會在你選的這一天進行。在比賽之前，勤勞的你每天都會訓練你的每一匹馬；而你的對手自我感覺非常良好，因此不會訓練他的馬。每一匹馬的素質不同，我們用 a_i 來表示第 i 匹馬的速度。每經過一天的訓練，你的每一匹馬的速度會增加 1。

現在你有你和對手共 $2N$ 匹馬的資料，請決定訓練的天數 M ，使得在第 $M+1$ 天比賽的時候，你有一個出場順序可以贏得 N 場比賽中的至少 K 場（不包含平手）。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 30$)，代表接下來有幾組測試資料。

每一組測試資料有三行。第一行有兩個數字， N 和 K 。

第二行有 N 個整數，第 i 個數字代表你的第 i 匹馬的速度 a_i ，兩個數字間以一個空白隔開。

第三行也有 N 個整數，第 j 個數字代表對手的第 j 匹馬的速度 c_j ，兩個數字間也以一個空白隔開。（ $1 \leq K \leq N \leq 100000$, $0 \leq a_i, c_j \leq 100000000$ ）

■ 輸出檔說明

對每筆測試資料輸出一個非負整數 M ，代表訓練 M 天後在第 $M + 1$ 天舉行賽馬你可以贏得至少 K 場。如果有不只一個 M 滿足條件，請輸出最小的 M 。

■ 範例輸入

```
2
3 2
3 1 2
0 4 3
1 1
1
10
```

■ 範例輸出

```
1
10
```

題目 G

數據加密

執行時間限制: 10 秒

在不安全的情形之下進行通訊是相當危險的。於是將要傳達的訊息加密是一件很重要的事情。讓我們考慮以下這種加密/解密方法，不妨稱之為「混亂加密/解密法」。

基本的概念，不外乎就是將想要傳遞的訊息以自己喜歡的順序任意重新排列，我們稱之為「亂序加密法」。例如我們將

`dogdogcatcatmeow`

重新排列之後可以得到

`acgtdowecmoodtag`

於是攔截訊息的人會沒辦法一下子猜出答案。不過這也會讓收到訊息的人很困擾，往往會因為不知道要怎麼找出正確的排列，還原出原本的訊息而感到徬徨。

這時候，另一種加密方法：隱藏字串加密法出現了！我們只要將想要的訊息隱藏進去就可以了，例如

`helloworld`

加密後可以變成

`homelesslogwouldbearealdog`

換句話說，正確的明文一定是加密後密文的子序列。密文長度越長，則安全等級越高。不過，收到這樣加密訊息的人應該還是會覺得很困擾。

不如來結合以上兩種加密方法，從而得到「混亂加密法」吧！讓我們考慮“pleap;tampaple”這樣的密文，前半部是亂序加密法的產物，後半部則是隱藏字串加密法的結果。而解密出來的明文，其實就是同時符合兩者條件的、字典順序最小的字串“apple”。(註：雖然“papple”也符合亂序加密法和隱藏字串加密法的結果，但是它的字典順序比“apple”還要來得大。因此“papple”不是這個密文解回來的明文。)

現在給你一個待加密的明文 S ，和希望用混亂加密法加密出來的長度 n 。請找到以「混亂加密法」加密後，所有可能的字串中字典順序最小的密文。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 1000$)，代表接下來有幾組測試資料。

每一筆測試資料包含一個長度不超過 100 的字串 S ，以及一個整數 n ($2|S| + 1 \leq n \leq 1000$ ，其中 $|S|$ 為字串 S 的長度)。你可以假設所有輸入的字串僅包含小寫英文字母。

■ 輸出檔說明

對於每一筆測試資料，請輸出滿足題目要求的密文。若所要求的密文不存在，請輸出 NONE。

■ 範例輸入

```
5
apple 11
apple 12
apple 13
aba 14
abcba 15
```

■ 範例輸出

```
aelp;apple
aelp;aapple
aelp;aaapple
aab;abaaaaaaaa
aabbc;abcbaaaaa
```