

# 2019 網際網路程式設計全國大賽

## 高中組決賽

- 本次比賽共 8 題，含本封面共 18 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，以題目敘述為主。
- 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，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 Time Limit Exceeded 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差不超過  $\epsilon$  皆視為正確」， $\epsilon$  值以題目敘述為主。  
舉例來說，假設  $\epsilon = 10^{-6}$  且  $a$  是正確答案， $b$  是你的答案，如果符合  $\frac{|a-b|}{\max(|a|,|b|,1)} \leq 10^{-6}$ ，就會被評測程式視為正確。

|      | 題目名稱    | 時間限制 (秒) | 記憶體限制   |
|------|---------|----------|---------|
| 題目 A | 逆序輸出    | 5        | 1 MB    |
| 題目 B | 忙碌的國度   | 4        | 1024 MB |
| 題目 C | 最小三倍完全數 | 1        | 1024 MB |
| 題目 D | 回文樹     | 1        | 1024 MB |
| 題目 E | 密碼鎖     | 4        | 1024 MB |
| 題目 F | 猜數字     | 1        | 1024 MB |
| 題目 G | 航線建設    | 10       | 1024 MB |
| 題目 H | 鐵路維修問題  | 6        | 1024 MB |

# 2019 網際網路程式設計全國大賽

## 輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main()
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. 逆序輸出

Problem ID: reverse

殿王最近收到了新的禮物：一台神秘的機器。根據殿王的研究，他發現這台機器的功能類似計數器：這台機器有 24 個儲存單元，每個儲存單元都只能儲存一個整數。在剛開機的時候，所有儲存單元的值都會是 0。接著可以執行三種指令以操作這台機器：把某個儲存單元的值複製到另一個儲存單元、把某個儲存單元的值增加 1、輸出某個儲存單元的值。這台機器每執行一個指令要花恰好 1 秒的時間。

很電的殿王很快地發現這台機器雖然看起來每次只能加一，但是只要適當的操作這台機器，就可以讓這台機器「逆序輸出」，也就是任意給定一個正整數  $N$ ，殿王可以讓這台機器依序輸出  $N, N - 1, \dots, 1, 0$ ！

然而，殿王遠不滿足於此。他希望這台機器可以看起來像是能執行「減一」一樣，所以他希望這台機器在逆序輸出的同時，在兩次連續的輸出之間不能間隔超過 16 秒。也就是說，殿王希望這台機器在輸出  $N$  之後 16 秒以內就要輸出  $N - 1$ （相當於這兩次輸出的中間只能執行最多 15 個指令），輸出  $N - 1$  之後 16 秒以內就要輸出  $N - 2$ ，以此類推。另外，殿王也希望這台機器在輸出 0 之後可以立刻結束。（殿王只有要求兩次連續的輸出之間間隔，因此第一次輸出（也就是輸出  $N$ ）之前可以執行任意數量的指令。）

你，身為能進入 NPSC 的決賽選手，殿王決定把這個重要任務交給你。請你構造一系列的指令，以完成上述殿王的要求。

### Input

輸入只有一行包含一個正整數  $N$ ，意義如題目所述。

- $1 \leq N \leq 10^6$

### Output

請依序輸出所要執行的指令，每個指令一行，格式如下：

- $1\ x\ y$  ( $0 \leq x, y < 24$ )：把第  $y$  個儲存單元的值複製到第  $x$  個儲存單元。
- $2\ x$  ( $0 \leq x < 24$ )：把第  $x$  個儲存單元的值增加 1。
- $3\ x$  ( $0 \leq x < 24$ )：輸出第  $x$  個儲存單元的值。

輸出完所有指令後，請再輸出一行“0”（不含引號），代表輸出結束。

**Sample Input 1**

3

**Sample Output 1**

```
2 0
1 1 0
2 1
1 2 1
2 2
3 2
3 1
1 1 0
3 1
3 3
0
```

## B. 忙碌的國度

Problem ID: busy

逼逼國是個忙碌的國家，裡面的人們每天都工作到很晚，導致有些人下班之後很難找到還沒打烊的餐廳吃飯，所以現在逼逼國的市長 bb 決定請你幫忙大家規劃吃晚餐的行程。

逼逼國裡面有  $N$  間公司，第  $i$  間公司位在二維平面上  $(x_i, y_i)$  的位置，並且每天固定在  $t_i$  的時間下班。員工在公司下班的那刻就可以離開公司。

另外逼逼國裡面有  $M$  間餐廳，第  $i$  間餐廳位在二維平面上  $(p_i, q_i)$  的位置，每天固定在  $c_i$  的時間打烊，並且有著以正整數  $v_i$  表示的美味程度。顧客只要在時間點  $c_i$  以前（包含  $c_i$ ）抵達  $(p_i, q_i)$  就可以購買到這間餐廳販售的餐點。

在逼逼國裡車子只能沿著平行於  $X$  軸或是  $Y$  軸的方向行駛，並且每單位時間可以行駛一單位的距離。所以如果要透過車子從  $(a, b)$  行駛到  $(c, d)$  必須要花費  $|a - c| + |b - d|$  的時間。

綜合以上的資訊，如果在第  $i$  間公司上班的員工下班之後想要去第  $j$  間餐廳買晚餐，必須要滿足  $t_i + |p_i - x_i| + |q_i - y_i| \leq c_i$ 。

請問對於逼逼國裡每一間公司的員工，他下班之後能吃到的餐廳最大的美味程度可以多大呢？

### Input

輸入的第一行有兩個正整數  $N, M$ ，代表逼逼國的公司數量以及餐廳數量。

接下來  $N$  行，每一行代表一間公司。第  $i$  行會有三個正整數  $x_i, y_i, t_i$ ，代表著第  $i$  間公司的位置以及下班時間。

接下來  $M$  行，每一行代表一間餐廳。第  $i$  行會有四個正整數  $p_i, q_i, c_i, v_i$ ，代表著第  $i$  間餐廳的位置、關門時間以及美味程度。

- $1 \leq N, M \leq 10^5$
- $1 \leq x_i, y_i, p_i, q_i \leq 5 \times 10^8$
- $1 \leq t_i, c_i \leq 10^9$
- $1 \leq v_i \leq 10^6$

## Output

輸出有  $N$  行，第  $i$  行代表第  $i$  間公司的員工在下班之後能吃到的餐廳的最大美味程度，如果這間公司的員工在下班之後吃不到任何的餐廳請輸出 "-1"（不含雙引號）。

### Sample Input 1

```
3 2
2 2 3
5 4 9
3 5 14
2 6 15 8
6 4 9 15
```

### Sample Output 1

```
15
8
-1
```

### Sample Input 2

```
5 5
5 18 26
19 14 9
17 13 16
6 6 13
17 3 21
17 7 21 20
11 17 30 16
11 19 5 10
17 10 1 11
19 14 11 7
```

### Sample Output 2

```
-1
20
16
16
-1
```

## C. 最小三倍完全數

Problem ID: baisekikanzensuu

喜歡音樂遊戲的小 B 最近在瀏覽「いちかの BEMANI 投票選拔戰 2019」的參戰樂曲當中看到了一首歌的歌名寫作「最小三倍完全數」。於是充滿著好奇心的小 B 就上網查了「三倍完全數」到底是什麼東西。

如果一個數  $n$  是  $k$  倍完全數的話，那就代表將  $n$  的所有因數加總之後剛好是  $kn$ 。例如 28 是一個 2 倍完全數，因為  $1 + 2 + 4 + 7 + 14 + 28 = 56 = 2 \times 28$ 。而所謂的最小三倍完全數就是指所有的三倍完全數當中最小的那一個，也就是 120。

好奇的小 B 現在想知道這世界上有沒有比這些更高倍數的完全數，於是他前來向你求助，請告訴他是否存在一個正整數  $n_k$ ，使得  $n_k$  是  $k$  倍完全數。

### Input

輸入只有一行，包含一個整數  $k$ ，代表小 B 想知道  $n_k$ 。

- $1 \leq k \leq 5$

### Output

輸出一行，如果  $n_k$  存在，包含一個正整數  $n_k$ ，否則請輸出  $-1$ 。

保證如果對於某個  $k$ ， $n_k$  存在，那麼在  $10^{15}$  以下一定存在一個數，且該數是  $k$  倍完全數。

輸出的  $n_k$  不可以超過  $10^{15}$ 。

#### Sample Input 1

|   |
|---|
| 2 |
|---|

#### Sample Output 1

|   |
|---|
| 6 |
|---|

#### Sample Input 2

|   |
|---|
| 3 |
|---|

#### Sample Output 2

|     |
|-----|
| 120 |
|-----|

*This page is intentionally left blank.*



## D. 回文樹

Problem ID: palindrome

你聽過回文樹嗎？這是一種可以在  $O(N|\Sigma|)$  的時間內算出一個長度為  $N$  的字串  $S$  的所有回文子字串出現次數的資料結構，其中  $|\Sigma|$  為字元集的大小。例如，當  $S$  是 `abacaba` 時，子字串 `a`、`b`、`aba`、`aca`、`bacab`、`abacaba` 皆是回文，且其中 `aba` 出現了兩次、`a` 出現了四次。

由於回文樹不僅功能強大，種在路邊作為行道樹也可以美化市容。更重要的是，如果用回文來灌溉它的話，它就會將城市中的廢氣轉為清新的空氣！長年受到空氣污染影響的 NPSC 國當然不會放過這個機會，因此 NPSC 國政府從回文樹的原生地 CSPN 國購進了大量的回文樹，並打算用大量的回文來灌溉它。

不過，NPSC 國很快就發現了，如果灌溉用的回文的字典序不夠小的話，可能反而造成反效果，使得回文樹枯萎而死。因此，政府打算重新排列手上的字串，使得它變成回文且字典序最小。請幫助 NPSC 國找到字典序最小的回文，解決空汙的問題！

### Input

輸入第一行有一個正整數  $T$ ，代表回文樹的肥料個數。接著  $T$  行，每行有一個非空字串  $S$ ，代表一個灌溉用的字串。

- $1 \leq T \leq 100$
- 輸入字串的總長度不會超過  $10^6$ 。
- $S$  只由大寫英文字母組成。

### Output

輸出  $T$  行，每行代表該字串重新排列後能夠造成的最小回文。若不存在一種重排成回文的方式，輸出 `NPSCCSPN`。

| <b>Sample Input 1</b> | <b>Sample Output 1</b> |
|-----------------------|------------------------|
| 2<br>ACCBBA<br>ABC    | ABCCBA<br>NPSCCSPN     |

## E. 密碼鎖

Problem ID: lock

bb 是個珠寶蒐藏家，顧名思義，他有很多貴重的珠寶。為了好好保護這些珠寶，bb 請他的朋友小 B 幫他設計了一個保險櫃，並且裝上了密碼鎖。這個密碼鎖有  $N$  個轉盤，每個轉盤上有  $N$  個刻度，只有當  $N$  個轉盤的刻度都轉到正確的位置時，才能打開保險櫃。不過，由於機械設計上的限制，任兩個轉盤不能轉到同一個刻度，例如當  $N = 3$  時，1, 1, 3 這組密碼是不可能被輸入的。

由於 bb 很健忘，所以他常常忘記他設定的密碼。因此他請小 B 額外幫他設計一個提示功能，每當 bb 輸錯密碼時，這個提示功能會顯示當前 bb 輸入的密碼「最少」還需要轉幾次轉盤才能得到正確的密碼（這裡轉一次轉盤的定義是將任意一個轉盤順時鐘或是逆時鐘旋轉一個刻度）。不過，由於小 B 找不到可以顯示多位數的顯示器，所以這個提示功能只會顯示個位數。

bb 除了喜歡蒐藏珠寶外，數學也是他的嗜好之一。因此他很好奇，對於一個整數  $K$ ，總共有幾組合法的輸入使得提示功能會顯示  $K$  呢？

雖然 bb 很會寫程式，可是他不想寫。所以請你寫一個程式來回答 bb 的問題。

### Input

輸入第一行有一個正整數  $T$ ，代表測試資料的數量。接著有  $T$  筆測試資料，每一筆有兩行。第一行有兩個整數  $N$  與  $K$ ，代表 bb 密碼的長度以及 bb 的問題。下一行有  $N$  個正整數，代表 bb 的密碼。

- $1 \leq T \leq 10^4$
- $1 \leq N \leq 20$
- $0 \leq K \leq 9$
- 保證 bb 的密碼是一組合法的輸入。

### Output

對於每一筆測試資料，輸出一行代表答案。由於答案可能很大，請輸出其除以  $10^9 + 7$  的餘數。

| Sample Input 1  | Sample Output 1            |
|---|----------------------------|
| 5<br>3 0<br>1 2 3<br>3 1<br>1 2 3<br>3 2<br>1 2 3<br>3 3<br>1 2 3<br>10 4<br>7 1 2 3 5 8 9 4 6 10 | 1<br>0<br>3<br>2<br>729800 |

## F. 猜數字

Problem ID: guess

「現在有一個介於 1 到  $N$  之間（含）的數字  $x$ ，你不知道它是多少，但是你每次可以詢問一個正整數  $y$ ，系統將會回答  $y$  是不是小於  $x$ ，請你用最少的詢問次數找出  $x$  是多少。」

以上是一個在程式練習當中時常出現的基礎題，相信進入決賽的你一定有看過吧！那麼現在來試試一題進階一點的題目吧！

現在有  $K$  個介於 1 到  $N$  之間（含）的數字  $x_1, x_2, \dots, x_K$ ，你不知道它們分別是多少，但是你每次可以詢問一個正整數  $y$ ，系統將會對每個  $i$  回答  $y$  是不是小於  $x_i$ ，請你用最少的詢問次數找出  $x_1, x_2, \dots, x_K$  分別是多少。

### 互動說明

首先，請由標準輸入 (standard input) 輸入兩個以空白隔開的正整數  $N, K$ ，意義如題目所述。

當你想要詢問一個數字  $y$ ，請輸出一行包含  $y$ ，這個整數必須介於 1 到  $N$  之間。當你輸出完這個數字之後，記得要清空 (flush) 標準輸出 (standard output)。

當系統收到這個詢問之後，會回覆一行到你的標準輸入，這一行包含  $K$  個以空白隔開的字串，每個字串都是 "Yes" 或 "No"（不含引號）兩者之一，如果第  $i$  個字串是 "Yes"，代表  $y < x_i$ ，否則代表  $y \geq x_i$ 。

當你確定了  $x_1, x_2, \dots, x_K$  的值，請你先輸出一行包含一個 0。接下來，請你輸出  $K$  行，每行有一個數字，其中第  $i$  行的數字代表你認為的  $x_i$ 。輸出完這些整數後，你的程式必須立刻結束 (exit)。

- $1 \leq N \leq 10^{18}$
- $1 \leq K \leq 200$
- 你的程式必須使用最少的詢問次數找出答案。也就是說，評測程式在每一次回答詢問之後會計算在當前給定的條件之下，若使用最佳的詢問策略，最多需要幾次詢問才能確定答案。如果你的程式最後的詢問次數超過這個數值，你的程式將會被強制結束，且被視為輸出了錯誤的答案 (Wrong Answer)。

下列兩段範例程式分別以 C 語言、C++ 語言撰寫。這個範例程式讀入  $N, K$ ，詢問 1，如果  $1 < x_i$  就回報  $x_i = N$ ，否則回報  $x_i = 1$ 。

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     int N, K, i;
5     char str[5];
6     int ans[200];
7     scanf("%d%d", &N, &K);
8     printf("%d\n", 1);
9     fflush(stdout);
10    for (i = 0; i < K; i++) {
11        scanf("%s", str);
12        ans[i] = (strcmp(str, "Yes") == 0);
13    }
14    puts("0");
15    for (i = 0; i < K; i++) {
16        printf("%d\n", ans[i] ? N : 1);
17    }
18 }
```

```
1 #include <string>
2 #include <iostream>
3 int main() {
4     int N, K;
5     std::string str;
6     bool ans[200];
7     std::cin >> N >> K;
8     std::cout << 1 << '\n' << std::flush;
9     for (int i = 0; i < K; i++) {
10        std::cin >> str;
11        ans[i] = (str == "Yes");
12    }
13    std::cout << "0\n";
14    for (int i = 0; i < K; i++) {
15        std::cout << (ans[i] ? N : 1) << '\n';
16    }
17 }
```

## G. 航線建設

Problem ID: flight

貓貓星球是個充滿貓貓的星球，星球上的每隻貓貓都十分的可愛，令人看了都忍不住想要對著貓貓們喵喵叫呢！

俗話說的好，貓兒當自強！貓貓星球中的各個城市其實距離都蠻遠的，需要由一些飛機航線來連結各個城市的交通。

貓貓國共有  $N$  個城市，編號為  $1, 2, \dots, N$ ，並且有一些航線連接著兩城市，每條航線都會有其飛行所需的費用。除了飛行外，沒有其他方式可以在兩城市間穿梭。

眾所皆知，在某些情況下，空姐貓貓能會藉由罷工來爭取自己的權利，因此有些航線可能會因此停駛。同時，也可能會有新航線的出現。

與地球相似地，貓貓星球要在空中飛行必須要獲得飛行許可證，而在該星球中，發放該許可證的是貓貓星球的主宰 — 殿王。殿王為了使貓貓星球維持良好的物價環境，因此殿王會選擇當前最低總費用的航班組合來頒發飛行許可證，使得任兩個城市都可以藉由這些航班組合，直接或間接的連接。

由於新航班與航班關閉的事件每天都在發生，殿王必須隨時地來更新飛行許可證，但由於航班的組合太多了，殿王無法很快地找出最佳的組合，因此殿王想請你幫她找出來這樣的組合。為了方便驗證，你只要回答最優組合中連接兩城市的最多航班數是多少。

### Input

輸入第一行有兩個整數  $N, Q$ ，代表貓貓星球的城市數量以及事件數。

接下來的  $Q$  行，每行有四個整數  $q_i, u_i, v_i, c_i$ ，用來表示一個事件，形式為以下兩種中的一種：

- $0 u_i v_i c_i$ ：代表一條連接著  $u_i$  和  $v_i$  城市且飛行費用為  $c_i$  的航線關閉了（該操作保證航線存在）。
- $1 u_i v_i c_i$ ：代表出現了一條連接著  $u_i$  和  $v_i$  城市且飛行費用為  $c_i$  的航線（保證該費用與當前其他航線的費用皆相異）。

- $1 \leq N \leq 10000$
- $1 \leq Q \leq 20000$
- $0 \leq q_i \leq 1$
- $1 \leq u_i \neq v_i \leq N$
- $0 \leq c_i \leq 10^9$

## Output

輸出共  $Q$  行，對於每個事件，輸出最優組合中連接兩城市的最多航班數是多少。如果存在兩城市無法藉由當前的航線到達，請輸出 -1。

### Sample Input 1

```
3 6
1 1 2 1
1 1 3 2
1 2 3 4
1 1 2 8
0 1 2 1
0 2 3 4
```

### Sample Output 1

```
-1
2
2
2
2
2
2
```



## H. 鐵路維修問題

Problem ID: path

NPSC 國由  $N$  個城市組成，分別編號為 1 號到  $N$  號。為了促進城市與城市之間的交流，政府決定建造  $N - 1$  條道路，其中第  $i$  條道路連接編號  $u_i$  的城市以及編號  $v_i$  的城市，且任一個城市一定能透過一些道路到達其他所有的城市。此外，由於各城市的路況不一，每條道路的「困難度」也不盡相同。其中，第  $i$  條道路有著困難度  $w_i$ （困難度數值越高的路越崎嶇、越難行駛）。

當然，為了使得居民可以在城市之間自由通行，NPSC 國也需要建造鐵路系統。此外，由於 NPSC 國的國民生性懶惰，為了讓乘客們都不需要換車，鐵路局決定在任兩個城市之間都規劃專車。也就是說，對於任兩個城市  $x \neq y$ ，都有一台班車從  $x$  開始，走過最少的道路到達  $y$ ，且一天只有一班。每台車在經過一天的載客之後都要進行維修，而維修的費用等於該台車「行進的距離」乘以「路上最崎嶇道路的困難度」，其中行進的距離就是從起點到終點所經過的道路數量。由於 NPSC 國的預算有限，因此請你先幫 NPSC 國計算一下每一天總共需要花多少錢維修。

### Input

輸入第一行有一個正整數  $N$ ，代表 NPSC 國的城市數量。接著  $N - 1$  行，第  $i$  行有三個正整數  $u_i, v_i, w_i$ ，代表一條連接城市  $u_i$  以及城市  $v_i$  的道路，其困難度為  $w_i$ 。

- $1 \leq N \leq 10^5$
- $1 \leq u_i, v_i \leq N$
- $1 \leq w_i \leq 10^9$
- 保證從任一個城市開始，一定能經過一些道路達到任意城市。

### Output

輸出一個整數，代表一天的總維修費用。由於答案可能很大，請輸出其除以  $10^9 + 7$  的餘數。

| Sample Input 1                        | Sample Output 1 |
|---------------------------------------|-----------------|
| 5<br>1 2 1<br>2 3 2<br>3 4 3<br>4 5 4 | 130             |

| Sample Input 2                                 | Sample Output 2 |
|--|-----------------|
| 6<br>1 2 4<br>3 1 3<br>5 3 2<br>5 6 5<br>6 4 4 | 318             |

| Sample Input 3   | Sample Output 3 |
|--|-----------------|
| 15<br>4 11 1<br>1 3 3<br>6 12 9<br>5 14 7<br>6 15 1<br>8 7 7<br>5 8 2<br>5 9 2<br>14 10 1<br>5 2 2<br>7 13 9<br>15 4 9<br>9 1 5<br>6 5 9 | 5630            |