

# 一百學年度全國高級中學資訊學科能力競賽決賽

## 上機程式設計題

### 作答注意事項：

- 一、 對考題有任何疑義，請於考試開始後二個小時之內填寫「問題單」，交付監考人員轉送命題委員提出問題，逾時不予回覆。
- 二、 第一題和第二題每題 20 分，第三題到第六題各 15 分，共 100 分。
- 三、 可選擇指定解題語言中任何一種語言解題。
- 四、 最後繳交編譯後之執行檔限定在 Windows XP 的命令提示字元下執行。
- 五、 各題執行檔檔名請設定如下：  
考生編號\_題號.exe  
例如：101\_1.exe
- 六、 各題原始碼檔名請設定如下：  
考生編號\_題號.解題語言附屬檔名  
例如：101\_1.c
- 七、 各題輸入資料檔名如下：  
in\_題號.txt  
例如：in\_1.txt
- 八、 各題輸入方式以讀檔方式為之，請以目前工作目錄（current working directory）為讀取路徑。
- 九、 各題輸出方式為標準輸出（螢幕）。
- 十、 考生應隨時備份，以防資料流失。隨身碟之備份格式如下：以考生編號加題號為各題的目錄名稱；該目錄下至少須存放該題的執行檔及原始碼檔。  
例如：在隨身碟 \101\_1 這個目錄下，儲存 101\_1.c 和 101\_1.exe 這兩個檔案。
- 十一、 考試結束後，將不再允許更動及重新編譯程式。
- 十二、 所有發展的程式必須在 2 秒以內於試場內的電腦輸出結果，否則不予計分。

# 1. 搬雕像

時間限制：2 秒

## 問題敘述：

王先生是一位收藏家，他收集了非常多有名的雕像。某天，為了美觀，他想要將收藏台上的雕像按照某種方式重新擺放，由於雕像都有一定的重量，所以他決定雇用一位年輕人，小明，來幫忙搬雕像。

王先生收藏的雕像目前是以隨機的方式放在收藏台上，收藏台的位置由左至右成一系列排開，編號依序為 1, 2,..., n，每個收藏台上放置一個雕像，而收藏台  $i$  ( $1 \leq i \leq n$ ) 上目前放的雕像編號為  $S_i$ ，其高度為  $h_i$  公分，重量為  $w_i$  公斤。王先生要求小明依照下列方式去重新擺放雕像：

- (a) 搬動過程中，一次只能搬一個雕像，而每個收藏台可暫時放置 0 個、1 個、或多個雕像。
- (b) 完成重新擺放之後需符合下列條件：
  - 每個收藏台上放置一個雕像。
  - 雕像必須根據高度，由低至高從最左邊的收藏台開始依序放置。
  - 若任二雕像的高度相同時，則重量輕的雕像放置在左邊。
  - 若任二雕像高度和重量都相同時，則依照原先雕像的左右相對順序來放置，也就是說原先在左邊的雕像必須放置在左邊。

為了節省搬運的距離，小明希望你替他寫出一個程式，根據上述方式將雕像重新擺放在收藏台上且搬動的總距離為最短。本題假設任二相鄰收藏台的距離為 1 公尺。

以下為一個範例，假設有 5 個收藏台，雕像  $S_i$  ( $1 \leq i \leq 5$ ) 的高度和重量以  $(h_i, w_i)$  表示，並依序為  $(5,20)$ ， $(10,25)$ ， $(78,40)$ ， $(25,25)$ ， $(5,15)$ 。一種搬動方式如圖(a)所示，搬動的總距離為 12 公尺，而另一種搬動方式如圖(b)所示，搬動的總距離為 8 公尺，是所有符合搬動方式中的最短距離。

收藏台	1	2	3	4	5
雕像	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>

將 S<sub>2</sub> 搬到收藏台 5，距離為 3 (公尺)

收藏台	1	2	3	4	5
雕像	S <sub>1</sub>		S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub> S <sub>2</sub>

將 S<sub>5</sub> 搬到收藏台 2，距離為 3 (公尺)

收藏台	1	2	3	4	5
雕像	S <sub>1</sub>	S <sub>5</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>2</sub>

將 S<sub>1</sub> 搬到收藏台 2，距離為 1 (公尺)

收藏台	1	2	3	4	5
雕像		S <sub>5</sub> S <sub>1</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>2</sub>

將 S<sub>2</sub> 搬到收藏台 3，距離為 2 (公尺)

收藏台	1	2	3	4	5
雕像		S <sub>5</sub> S <sub>1</sub>	S <sub>3</sub> S <sub>2</sub>	S <sub>4</sub>	

將 S<sub>3</sub> 搬到收藏台 5，距離為 2 (公尺)

收藏台	1	2	3	4	5
雕像		S <sub>5</sub> S <sub>1</sub>	S <sub>2</sub>	S <sub>4</sub>	S <sub>3</sub>

將 S<sub>5</sub> 搬到收藏台 1，距離為 1 (公尺)

收藏台	1	2	3	4	5
雕像	S <sub>5</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>4</sub>	S <sub>3</sub>

總距離: 3 + 3 + 1 + 2 + 2 + 1 = 12 (公尺)

圖(a)

收藏台	1	2	3	4	5
雕像	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>

將 S<sub>1</sub> 搬到收藏台 2，距離為 1 (公尺)

收藏台	1	2	3	4	5
雕像		S <sub>2</sub> S <sub>1</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>

將 S<sub>2</sub> 搬到收藏台 3，距離為 1 (公尺)

收藏台	1	2	3	4	5
雕像		S <sub>1</sub>	S <sub>3</sub> S <sub>2</sub>	S <sub>4</sub>	S <sub>5</sub>

將 S<sub>3</sub> 搬到收藏台 5，距離為 2 (公尺)

收藏台	1	2	3	4	5
雕像		S <sub>1</sub>	S <sub>2</sub>	S <sub>4</sub>	S <sub>5</sub> S <sub>3</sub>

將 S<sub>5</sub> 搬到收藏台 1，距離為 4 (公尺)

收藏台	1	2	3	4	5
雕像	S <sub>5</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>4</sub>	S <sub>3</sub>

總距離: 1 + 1 + 2 + 4 = 8 (公尺)

圖(b)

### 輸入說明:

第一行有一個整數  $n$ ， $1 \leq n \leq 10000$ ，代表收藏台的個數。接下來的  $n$  行，每行有兩個整數以空白隔開，其中第  $i$  行 ( $1 \leq i \leq n$ ) 為目前放在收藏台  $i$  上雕像  $S_i$  的高度  $h_i$  (單位為公分) 和重量  $w_i$  (單位為公斤)， $h_i$  和  $w_i$  都介於 1 和 65536 之間。

### 輸出說明:

輸出一個整數，代表搬動雕像所需的最短總距離(單位為公尺)。

### 輸入範例 1:

```
5
5 20
10 25
78 40
25 25
5 15
```

### 輸出範例 1:

```
8
```

### 輸入範例 2:

```
8
5 15
3 5
9 13
13 20
24 30
40 50
9 12
5 15
```

### 輸出範例 2:

```
18
```

## 2. 神秘的進位問題

時間限制：2 秒

### 問題敘述：

在一個神秘的國家，他們有不同的文明，他們所使用的數字表示法跟常見的十進位法不一樣。對於一個十進位的數字  $N$ ，他們會表示成  $abc$ ，其中  $a > b > c \geq 0$ ，且滿足  $N = C(a, 3) + C(b, 2) + C(c, 1)$ ， $C$  為二項係數，即  $C(m, n) = m! / (n!(m-n)!)$ ，但當  $m < n$  時， $C(m, n) = 0$ 。為幫助了解這個神秘國度的文化，請寫一程式來將十進位數轉換成這個神秘的進位法。

### 條件說明：

1. 所有數字皆為非負整數。
2.  $0 \leq N \leq 500$ 。

### 輸入說明：

第一行有一個整數  $m$ ， $1 \leq m \leq 10$ ，代表要轉換的十進位數的個數。接下來的  $m$  行（第 2 行至第  $(m+1)$  行）：每一行都有一個介於 0 和 500 之間的整數，代表要轉換的十進位數。

### 輸出說明：

針對每一個十進位數分別在一行輸出對應的  $abc$ ，其間不需留空白，注意  $a$ 、 $b$  和  $c$  未必是只有一位數，若答案不唯一時請輸出字典順序最小的表示法，即盡可能取小的  $a$  及  $b$  值。

### 輸入範例 1：

```
4
0
1
2
200
```

**輸出範例 1：**

210  
310  
320  
1187

**輸入範例 2：**

3  
18  
19  
20

**輸出範例 2：**

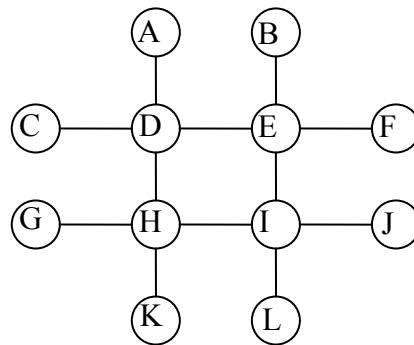
542  
543  
610

### 3. 新井字遊戲

時間限制：2 秒

#### 問題敘述：

在台灣宜蘭的一個小鄉鎮裡，小學生之間流傳著一個稱為『新井字』的雙人對弈遊戲。這種遊戲使用的道具是一個『井』字形的棋盤(如圖一所示)，和 12 個棋子。遊戲開始時，棋盤上每一個標示英文字母的地方，都可以隨機決定要不要擺放一個棋子(但是當然不能整個棋盤都不放棋子)。遊戲進行的規則是由兩個遊戲者輪流取走棋盤上的棋子，每人每次必須取走棋盤上任意 1 個或 2 個相鄰(棋盤上有直線相連)的棋子，而被迫取走最後一個棋子的人算輸，另一位遊戲者則得勝。



圖一

其實如果仔細思考，可以證明不管遊戲開始時的棋盤狀態(簡稱起始狀態)如何，先開始取棋子的人(先手)與其對手(後手)二人中必定恰有一人，只要他每次都用對自己最有利的方，最後一定會得勝。舉例來說，若起始狀態中只有 A、B、F 處有棋子，則先手不管取走哪一個棋子，後手只要再移除一個棋子，就可以逼迫先手去取最後一個棋子，因此後手有百分之百得勝的把握，我們可以說『這個起始狀態是後手有利』。再舉另一個例子，若起始狀態只有 A、C、D 處有棋子，則先手只要懂得先取走 A 與 D 處相鄰的兩個棋子，就可以迫使後手去取最後一個棋子，因此先手就有百分之百得勝的把握，我們可以說『這個起始狀態是先手有利』。本題就是要請你寫一個程式，來判斷在不同的起始狀態下到底是先手還是後手有利。

#### 輸入說明：

第一行只含一個整數  $n$  ( $1 \leq n \leq 10$ )，告訴你總共有多少個起始狀態需要判斷。接下來的  $n$  行，每一行表示一種起始狀態。起始狀態是以連續 12 個 1 或 0 的數字來依序表示 A 到 L 的位置上有沒有棋子，1 代表該位置有棋子，0 表示該

位置沒有棋子。

### 輸出說明：

對於輸入的每一種起始狀態，若該起始狀態為對先手有利，你的程式必須輸出一個字元 1，若對後手有利，則必須輸出一個字元 0。因為輸入總共有  $n$  個起始狀態，因此總共必須輸出  $n$  個字元在同一行，且這些字元之間不需留空白。

### 輸入範例 1：

```
1
110001000000
```

### 輸出範例 1：

```
0
```

### 輸入範例 2：

```
2
101100000000
001100110010
```

### 輸出範例 2：

```
11
```



## 4. 捷運路線

時間限制：2 秒

### 問題敘述：

隨著捷運線的增加，台北市捷運局決定開發一個 App，來提供大眾更便利的捷運搭乘經驗。這個 App 需要在任何時間點，找出給定之兩個捷運站之間，不含等車時間的最短搭車時間。所有捷運線都在 06:00 從第一站與最後一站同時列車進站對開，每隔五分鐘就發一班車，最後一班車則在 23:55 發車。捷運到站後（包含起站）停留時間都是一分鐘（所以第一班車其實是在 06:01 離站）。為了確保 App 的實用性，使用者抵達某捷運站時間必須早於該站捷運開車時間，才能順利的搭上捷運。舉例而言，若抵達時間為 13:55，而欲搭的捷運於 13:54 抵達，且將於 13:55 開車，那麼本班車就趕不上，必須等待下一班車。但若是 13:54 抵達捷運站，則立刻搭上本班捷運（雖然要一分鐘後捷運才會離站）。

### 輸入說明：

第一行有兩個正整數  $n, m$  ( $1 \leq n \leq 10, 1 \leq m \leq 100$ )，以空白隔開，代表共有  $n$  條捷運線，且有  $m$  個交叉點，每個交叉點代表兩條捷運線共站，本題不會有三條或三條以上的捷運線共站。接下來的  $n$  行，每行都有四個以上的正整數  $i, k, s_1, s_2, \dots, s_k$ ，連續兩個數字之間以空白隔開，其中  $i$  ( $1 \leq i \leq 100$ ) 為捷運線代號， $k$  ( $2 \leq k \leq 20$ ) 為該捷運線總站數， $s_p$  ( $1 \leq p \leq k$ ) 代表從  $(p-1)$  號站到  $p$  號站所需的行車時間(以分鐘為單位)，因為  $s_1$  是起站，因此  $s_1$  一定是 0。接下來有  $m$  行，每行四個正整數  $i, p, j, q$ ，連續兩個數字之間以空白隔開，代表  $i$  號捷運線的  $p$  號站與  $j$  號捷運線的  $q$  號站共站。最後有五行的使用者測試資料，每行有六個數字： $hh, mm, i, p, j, q$ ，連續兩個數字之間以空白隔開，代表 App 使用者可於  $hh:mm$  ( $6 \leq hh \leq 23, 0 \leq mm \leq 59$ ) 前抵達  $i$  號捷運線的  $p$  號站且希望前往  $j$  號捷運線的  $q$  號站。

### 輸出說明：

每個使用者測試資料都有一行的輸出，共五行。每行一個正整數，代表抵達目的地捷運站所需搭乘捷運最短時間(以分鐘為單位)，此時間不包含初始等待捷運或轉搭另一捷運線時在月台等待捷運進站時間。所有的測試資料都會在捷運停駛前可以抵達目的地。

以輸入範例(參閱下一頁)第一個測資：6 0 10 1 10 3 為例，使用者於 06:00 抵達 10 號捷運線的 1 號站，目的地為 10 號線的 3 號站，因此搭乘捷運所需的最短

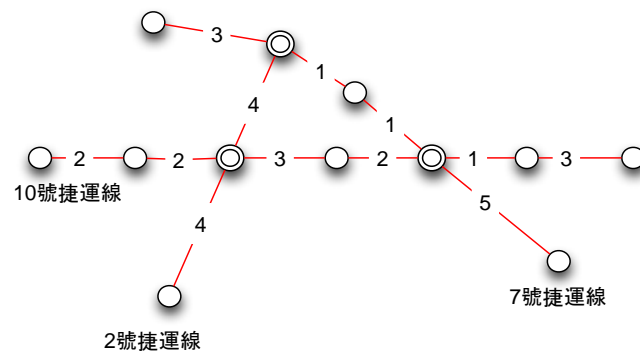
時間為：1 分鐘 (在捷運上等待離站) + 2 分鐘 (到達 2 號站所需時間) + 1 分鐘 (在捷運上等待離站) + 2 分鐘 (到達 3 號站所需時間) = 6 分鐘。若以輸入範例第四個測資：12 7 2 1 10 1 為例，使用者於 12:07 抵達 2 號捷運線的 1 號站，目的地為 10 號線的 1 號站，因此搭乘捷運所需的最短時間為：1 分鐘 (12:10 之捷運進站後搭上捷運並在捷運上等待離站) + 4 分鐘 (到達 2 號站所需時間) + 1 分鐘 (2 號線於 12:15 到站後下車等待 12:18 抵達的 10 號線，上車並在捷運上等待離站) + 2 分鐘 (到達 10 號線 2 號站所需時間) + 1 分鐘 (在捷運上等待離站) + 2 分鐘 (到達 10 號線 1 號站所需時間) = 11 分鐘。

### 輸入範例：

```

3 3
10 7 0 2 2 3 2 1 3
2 3 0 4 4
7 5 0 5 1 1 3
10 3 2 2
7 2 10 5
7 4 2 3
6 0 10 1 10 3
6 12 10 7 10 1
13 55 2 1 2 3
12 7 2 1 10 1
23 0 10 5 2 3

```



### 輸出範例：

```

6
19
10
11
4

```

## 5. 股票趨勢

時間限制: 2 秒

### 問題敘述：

彭先生任職於證券公司，是一位股票分析師。公司經理認為目前的股票分析軟體仍可再改進，希望彭先生再設計一套更準確的軟體。近日來，彭先生埋頭鑽研，他發現過去的研究結果，有人提到，如果能在歷史資料中，找到與近期股票走勢相近的樣型，即可使用此歷史樣型的交易策略，做為近期的買賣策略。為了驗證這樣的講法是否正確，彭先生從股票歷史資料抽出一些特徵資料，並以大寫英文字母 A~Z 代表特徵資料，因此股票資料變成一串的英文字母序列。判斷近期股票資料與某一段歷史資料是否相近，就變成判斷二串字母序列(長度不一定相等)的相似度，亦即找出兩者的最長共同子序列(LCS, longest common subsequence)。

在計算二串股票資料序列的相似度時，還有一個限制，兩個相似點(相同字母)的前後間距不能太遠，否則相似度會被扭曲。發現了這個特性後，彭先生將此問題正式定義為「有間距限制的最長共同子序列」(GLCS, gapped longest common subsequence)問題。

假設第一個序列稱為 $\alpha$ ，第二個序列稱為 $\beta$ 。例如， $\alpha$ ="ACBDCAA"， $\beta$ ="ADDBCDBAC"。兩者在無間距限制的情形下，其 LCS 可為"ADCA"，"ABCA"，或"ACBC"，長度為 4。假設間距限制如下：

A 2, B 0, C 3, D 0

上述間距之意義為，如果字母 A 被選入 LCS 中，則與其前一個被選入的字母之間，在 $\alpha$ 序列最多只能有 2 個未被選入的字母，在 $\beta$ 序列亦同。 $\alpha$ 與 $\beta$ 在上述間距限制的情形，GLCS 可為"ACA"或"ACC"，長度為 3。

對於無間距限制的情形，可將每個字母的間距視為無限大。本題的答案只要輸出 GLCS 的長度即可。

### 輸入說明：

共分成二部分。第一部分，第一行為 $\alpha$ 序列，第二行為 $\beta$ 序列，兩者都是大寫英文字母 A~Z 的序列，每個序列長度至少為 1，最長為 800。第二部分自第三行起，第三行有一個數值 k，代表以下有 k 組測試資料( $1 \leq k \leq 5$ )，每一組測試資料為一行，每一行有多個(可能零個)字母間距限制，每個限制的第一個為英文字母，第二個為間距數值(數值介於 0 與 400 之間)；英文字母不一定按照順序，也不一定每個字母都會出現，未出現的字母間距可視為無限大。每一行字母間距限制的最後一個符號為\$，代表該行(該組)的資料結束。每一行的字母間距限制情

形，相鄰兩項資料之間均有一個空白隔開。

### 輸出說明：

對於每一組測試資料，輸出它的 GLCS 長度。輸出這 k 個值於一行，且相鄰兩個整數之間以一個空白隔開。

### 輸入範例 1：

```
ACBDCAA
ADDBCDBAC
2
$
A 2 B 0 C 3 D 0 $
```

### 輸出範例 1：

```
4 3
```

### 輸入範例 2：

```
ACBDCAA
ADDBCDBAC
1
C 4 A 6 $
```

### 輸出範例 2：

```
4
```

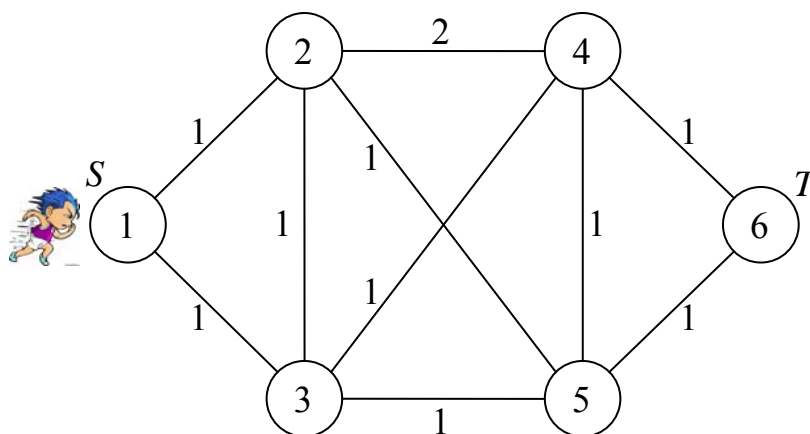
## 6. 城市馬拉松

時間限制：2 秒

### 問題敘述：

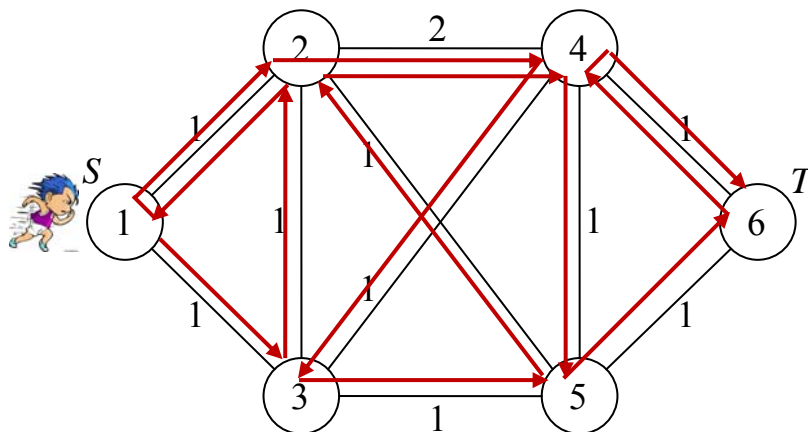
H城市舉辦馬拉松比賽。比賽區域有 $N$ 個補給站， $2 \leq N \leq 1000$ 。為了方便說明，我們將 $N$ 個補給站名稱以正整數 $1, 2, \dots, N$ 來表示。 $N$ 個補給站有街道來連接，使得選手可從任一個補給站出發，經由幾條街道抵達另一個補給站。我們可以用圖形來表示這些補給站跟街道之間的關係：節點表示補給站，而連接節點的連結線則代表連接兩個補給站之間的街道(如圖一所示，其中補給站名稱以圓圈內的數字來表示，而街道上的數字則代表跑完此街道所需花費的時間)。我們以符號 $(I, J)$ 來表示連接補給站 $I$ 和補給站 $J$ 的街道(連結線)。每一條街道 $(I, J)$ 都結合一個整數的權重值 $c(I, J)$ 來代表跑完 $(I, J)$ 這條街道所要花費的時間，其中 $c(I, J)$ 需滿足 $1 \leq c(I, J) \leq 999$ 。令  $N_{\text{odd}}$  代表那些與奇數條街道相接的補給站個數，則H城市有一個重要特性： $N_{\text{odd}}$  為偶數且  $0 \leq N_{\text{odd}} \leq 20$ 。

給定一個起點補給站 $S$ 和終點補給站 $T$  ( $S \neq T$ )，請寫一個程式計算選手從起始補給站 $S$ 出發，把每條街道都跑過至少一次且到達終點補給站 $T$ 所需花費的最短時間。注意：這個城市中所有的街道都是雙向道，同一個補給站和街道可被重複經過。

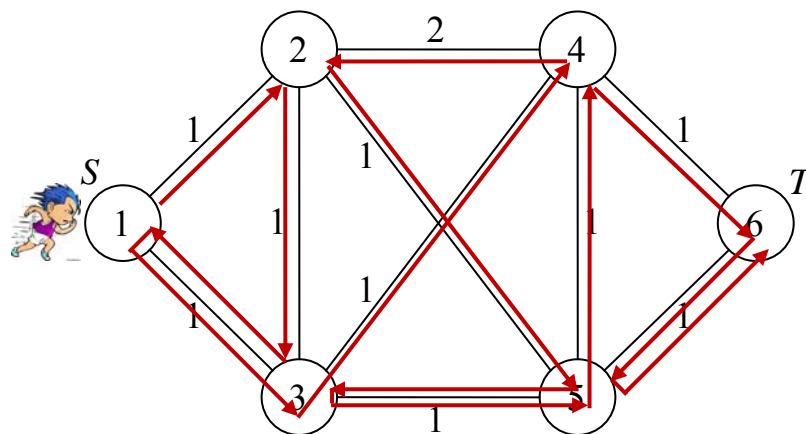


圖一

在圖一的例子中， $N_{\text{odd}} = 0$ ， $S=1$ ， $T=6$ 。圖二說明其中一種跑法為： $S=1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 6=T$ ，可在 15 單位時間從起點出發，將所有街道都跑過至少一次，且到達終點。然而此種跑法所需的時間並非最短。事實上，此例中花費時間為最短的跑法如圖三所示，為  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 6$ ，所花費時間為 14 單位。



圖二



圖三

### 輸入說明：

第一行有四個數字，連續兩個數字之間以空白符號做區隔。第一個數字  $N$  ( $2 \leq N \leq 1000$ ) 代表圖形的節點個數；第二個數字  $M$  ( $1 \leq M \leq N(N-1)/2$ ) 代表圖形的連結線個數；第三個數字則代表起點名稱；第四個數字則代表終點名稱。從第二行起連續有  $M$  行，表示  $M$  條連結線，每行有三個數字，連續兩個數字之間以空白符號做區隔：前二個數字代表連結線的兩個端點，第三個數字代表連結線的權重值。輸入保證任兩個補給站之間都有路徑相連， $N_{\text{odd}}$  為偶數且  $0 \leq N_{\text{odd}} \leq 20$ 。

### 輸出說明：

輸出一個整數，代表選手所花費的最短時間。

輸入範例 1：

```
6 10 1 6
1 2 1
1 3 1
2 3 1
2 4 2
2 5 1
3 4 1
3 5 1
4 5 1
4 6 1
5 6 1
```

輸出範例 1：

14

輸入範例 2：

```
3 3 1 2
1 2 4
1 3 6
2 3 5
```

輸出範例 2：

19