

I. 作者簡介

- A. 姓名：胡瑀真
- B. 系級：測量系 116 級
- C. 學號：F64126147
- D. e-mail：F64126147@gs.ncku.edu.tw

II. 程式簡介

A. 遊戲規則

遊戲背景為我方受到敵軍攻擊，我方和敵方皆以 4×4 的矩陣分配軍力，敵方的軍力分佈如預設、已分配完成，而玩家需擔任我方的指揮官，分配我方的軍力，分配的軍力總和不可超過我方原有的軍力，若超過則需重新分配（如：若預設我方共有 80 萬人之軍隊，則玩家不可在各個點位皆分配 6 萬大軍，因為 $6 \times 4 \times 4$ 等於 96 大於 80）。

玩家遵守規定並分配完成軍力後，我方和敵軍會開始交戰，我方軍力為矩陣 r ，敵方軍力為矩陣 s ，交戰規則為： $r[0][0]$ 和 $s[0][0]$ 對戰； $r[0][1]$ 和 $s[0][1]$ 對戰，以次類推至 $r[3][3]$ 和 $s[3][3]$ ，若我方在一點位的軍力大於或等於敵軍，則我方勝利，反之則我方失敗（如：若我方在 $r[0][1]$ 有 2 萬人的軍力，敵方在 $s[0][1]$ 同樣有 2 萬人的軍力，則我方在此點位會贏過敵方）。

在戰爭結束後，若我方贏過敵方的點位數量多於或等於敵方贏過我方的點位數量，則我方獲勝，反之我方失敗，若失敗，程式會印出我方在哪幾個點位失利、損失和倖存人數的情報，並開啟時光機，使時間倒流回戰爭開始之前，讓我方指揮官（玩家）依據情報重新分配軍力，至戰爭結束且我方勝利。

註：敵方分配的軍力預設為 $\{\{2,1,1,2\}, \{3,4,4,3\}, \{4,7,7,4\}, \{5,6,6,5\}\}$ 。

B. 編寫功能

先設定整數變數 a 依序至 q ，其中 a 至 p 表示 4×4 矩陣中的值， q 表示我方軍力總量，再設定兩整數矩陣 r 和 s ，其中 s 先預設好且為敵方軍力分佈。在輸出「我方受到敵軍進攻，請指揮官下達命令，部署軍力！」後，運用 `while` 迴圈，使判斷條件為 `true`，令迴圈只有遇到 `break` 或 `continue` 時會終止或從頭再來。

`while` 迴圈內，令四整數變數 x 、 y 、 win 、 $lose$ 皆為 0，其中 x 表我方

贏敵方的點位數量，y 表我方書敵方的點位數量，win 表倖存人數，lose 表損失人數。之後列印出 4*4 之矩陣模樣和我方軍力總量，並使玩家開始從 a 至 p 輸入我方軍力分佈。輸入完成後，運用 if-else 敘述判斷玩家輸入之軍力總和是否超過我方軍力總量，若超過，則印出「指揮官調動的軍力超出可供調動的軍力，請指揮官重新考慮！」並運用 continue 敘述使迴圈回到 while 迴圈的起點，重新分配兵力，若不超過，則進行 else 內敘述。

```

5  int main() {
6      int a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q=80,r[4][4];
7      int s[4][4]={2,1,1,2},{3,4,4,3},{4,7,7,4},{5,6,6,5}};
8      cout<<"我方受到敵軍進攻，請指揮官下達命令，部署軍力！"<<endl;
9      while (true){
10         int x=0,y=0,win=0,lose=0;
11         cout<<"a b c b"<<endl;
12         cout<<"e f g h"<<endl;
13         cout<<"i j k l"<<endl;
14         cout<<"m n o p"<<endl;
15         cout<<"我方共有"<<q<<"萬人的軍力可供指揮官調動！"<<endl;
16         cout<<"請輸入在 a 點部署的軍力（單位：萬人）："；
17         cin>>a;
18         cout<<"請輸入在 b 點部署的軍力（單位：萬人）："；
19         cin>>b;
20         cout<<"請輸入在 c 點部署的軍力（單位：萬人）："；
21         cin>>c;
22         cout<<"請輸入在 d 點部署的軍力（單位：萬人）："；
23         cin>>d;
24         cout<<"請輸入在 e 點部署的軍力（單位：萬人）："；
25         cin>>e;
26         cout<<"請輸入在 f 點部署的軍力（單位：萬人）："；
27         cin>>f;
28         cout<<"請輸入在 g 點部署的軍力（單位：萬人）："；
29         cin>>g;
30         cout<<"請輸入在 h 點部署的軍力（單位：萬人）："；
31         cin>>h;
32         cout<<"請輸入在 i 點部署的軍力（單位：萬人）："；
33         cin>>i;
34         cout<<"請輸入在 j 點部署的軍力（單位：萬人）："；
35         cin>>j;
36         cout<<"請輸入在 k 點部署的軍力（單位：萬人）："；
37         cin>>k;
38         cout<<"請輸入在 l 點部署的軍力（單位：萬人）："；
39         cin>>l;
40         cout<<"請輸入在 m 點部署的軍力（單位：萬人）："；
41         cin>>m;
42         cout<<"請輸入在 n 點部署的軍力（單位：萬人）："；
43         cin>>n;
44         cout<<"請輸入在 o 點部署的軍力（單位：萬人）："；
45         cin>>o;
46         cout<<"請輸入在 p 點部署的軍力（單位：萬人）："；
47         cin>>p;
48         if(q<a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p){
49             cout<<"指揮官調動的軍力超出可供調動的軍力，請指揮官重新考慮！"
<<endl;
50             continue;
51         }
52         else{

```

圖一、功能部分之程式碼 1

在 else 敘述中，先將玩家輸入的 a 至 p 的值，賦值為 r 矩陣，r[0][0] 為 a 之值，r[0][1] 為 b 之值，以次類推至 r[3][3] 為 p 之值。之後運用一個雙重 for 迴圈，判斷我方軍力和敵方軍力之大小關係。在第一層 for 迴圈中，令正整數 u 為 0，u 每次迴圈即加 1，至大於或等於 4 才停止迴圈，使迴圈運行 4 次。在第二層 for 迴圈中，令正整數 v 為 0，v 每次迴圈即加 1，至大於或等於 4 才停止迴圈，使迴圈運行 4 次。每次迴圈皆運用 if-else 敘述比較 r[u][v] 和 s[u][v] 之大小關係，若 r[u][v] 大於或等於 s[u][v]，則為我方在該點位的勝利，故使 x 加 1，win 加 r[u][v] 減 s[u][v] 之值；若 r[u][v] 小於 s[u][v]，則為我方在該點位的失利，故使 y 加 1，lose 加 r[u][v] 減 s[u][v] 之值。在雙重 for 迴圈執行結束後，開始執行 if-else 敘述，判斷我方是勝利或失利。

```

52  ▾      else{
53          r[0][0]=a;
54          r[0][1]=b;
55          r[0][2]=c;
56          r[0][3]=d;
57          r[1][0]=e;
58          r[1][1]=f;
59          r[1][2]=g;
60          r[1][3]=h;
61          r[2][0]=i;
62          r[2][1]=j;
63          r[2][2]=k;
64          r[2][3]=l;
65          r[3][0]=m;
66          r[3][1]=n;
67          r[3][2]=o;
68          r[3][3]=p;
69  ▾      for(int u=0;u<4;u++){
70  ▾          for(int v=0;v<4;v++){
71  ▾              if(r[u][v]>=s[u][v]){
72                  x++;
73                  win=win+r[u][v]-s[u][v];
74              }
75  ▾          else{
76              y++;
77              lose=lose+s[u][v]-r[u][v];
78          }
79      }
80      }

```

圖二、功能部分之程式碼 2

if-else 敘述中，若 x 大於或等於 y，則表示我方勝利，執行 if 敘述之內容：印出「恭喜指揮官勝利！」和用星字號排列而成的 WIN、3 個大拇指表示稱讚，再使用 break 敘述，離開 while 迴圈並結束遊戲。若 x 小於 y，即表示我方失利，執行 else 敘述之內容，除印出「敵軍陣列比我方強大，我方失利」外，也用 16 個 if 敘述比較 r[0][0]和 s[0][0]的大小，依序比較至 r[3][3]和 s[3][3]的大小，若 r 矩陣之值小於 s，即印出該點位的點位名（如：若 r[0][0]小於 s[0][0]，即印出 a），最後運用 lose 和 win 的值，分別印出我方損失和倖存人數，再印出「為獲取勝利，我方將開啟時光機使時間回到戰爭開始之前，請指揮官重新思考並調動軍力！」和「時光機已開啟！」並使用 continue 敘述，使迴圈回到 while 迴圈的起點。

```

81 ✓ if(x>=y){
82     cout<<"恭喜指揮官勝利!"<<endl;
83     cout<<" * * * *** * * "<<endl;
84     cout<<" * * * * * * * "<<endl;
85     cout<<" * * * * * * * "<<endl;
86     cout<<" * * * *** * * "<<endl<<endl;
87     cout<<" ** ** **"<<endl;
88     cout<<" *** ** **"<<endl;
89     cout<<" ***** ***** *****"<<endl;
90     cout<<" ***** ***** *****"<<endl;
91     cout<<" ***** ***** *****"<<endl;
92     cout<<" ***** ***** *****"<<endl;
93     break;
94 }
95 ✓ else{
96     cout<<"敵軍陣列比我方強大，我方失利"<<endl;
97     cout<<"在";
98 ✓ if(r[0][0]<s[0][0]){
99     cout<<"a ";
100 }
101 ✓ if(r[0][1]<s[0][1]){
102     cout<<"b ";
103 }
104 ✓ if(r[0][2]<s[0][2]){
105     cout<<"c ";
106 }
107 ✓ if(r[0][3]<s[0][3]){
108     cout<<"d ";
109 }
110 ✓ if(r[1][0]<s[1][0]){
111     cout<<"e ";
112 }
113 ✓ if(r[1][1]<s[1][1]){
114     cout<<"f ";
115 }
116 ✓ if(r[1][2]<s[1][2]){
117     cout<<"g ";
118 }
119 ✓ if(r[1][3]<s[1][3]){
120     cout<<"h ";
121 }
122 ✓ if(r[2][0]<s[2][0]){
123     cout<<"i ";

```

```

124     }
125     if(r[2][1]<s[2][1]){
126         cout<<"j ";
127     }
128     if(r[2][2]<s[2][2]){
129         cout<<"k ";
130     }
131     if(r[2][3]<s[2][3]){
132         cout<<"l ";
133     }
134     if(r[3][0]<s[3][0]){
135         cout<<"m ";
136     }
137     if(r[3][1]<s[3][1]){
138         cout<<"n ";
139     }
140     if(r[3][2]<s[3][2]){
141         cout<<"o ";
142     }
143     if(r[3][3]<s[3][3]){
144         cout<<"p ";
145     }
146     cout<<"點我方軍力不足！"<<endl;
147     cout<<"我方此次共損失"<<lose<<"萬人"<<endl;
148     cout<<"我方此次共倖存"<<win<<"萬人"<<endl;
149     cout<<"為獲取勝利，我方將開啟時光機使時間回到戰爭開始之前，請指
揮官重新思考並調動軍力！"<<endl;
150     cout<<"時光機已開啟！"<<endl;
151     continue;
152 }
153 }
154 }
155
156 return 0;
157 }

```

圖三、功能部分之程式碼 3

III. 結果簡介

A. 輸入格式與運作流程

輸入 a 後，按下 enter 鍵，再輸入 b 後，按下 enter 鍵，以次類推至輸入 p 後按下 enter 鍵，即可看見輸出畫面（如圖四）。

因使用 while 迴圈、if-else 敘述、break 敘述和 continue 敘述，若輸入不符合規則或失敗，則在看到輸出畫面的同時，看見輸入畫面並再次輸入。

```

我方受到敵軍進攻，請指揮官下達命令，部署軍力！
a b c b
e f g h
i j k l
m n o p
我方共有80萬人的軍力可供指揮官調動！
請輸入在 a 點部署的軍力（單位：萬人）：3
請輸入在 b 點部署的軍力（單位：萬人）：2
請輸入在 c 點部署的軍力（單位：萬人）：2
請輸入在 d 點部署的軍力（單位：萬人）：3
請輸入在 e 點部署的軍力（單位：萬人）：4
請輸入在 f 點部署的軍力（單位：萬人）：5
請輸入在 g 點部署的軍力（單位：萬人）：5
請輸入在 h 點部署的軍力（單位：萬人）：4
請輸入在 i 點部署的軍力（單位：萬人）：5
請輸入在 j 點部署的軍力（單位：萬人）：8
請輸入在 k 點部署的軍力（單位：萬人）：8
請輸入在 l 點部署的軍力（單位：萬人）：5
請輸入在 m 點部署的軍力（單位：萬人）：6
請輸入在 n 點部署的軍力（單位：萬人）：7
請輸入在 o 點部署的軍力（單位：萬人）：7
請輸入在 p 點部署的軍力（單位：萬人）：6

```

圖四、輸入格式

B. 結束畫面

共調動 $3+2+2+3+4+5+5+4+5+8+8+6+6+7+7+6=80$ 萬人之軍力，並未超過可供調動之軍力，故不用重新輸入分配軍力。又因每個點位的軍力皆大於敵軍 1 萬人，故勝利。

```

我方受到敵軍進攻，請指揮官下達命令，部署軍力！
a b c b
e f g h
i j k l
m n o p
我方共有80萬人的軍力可供指揮官調動！
請輸入在 a 點部署的軍力（單位：萬人）：3
請輸入在 b 點部署的軍力（單位：萬人）：2
請輸入在 c 點部署的軍力（單位：萬人）：2
請輸入在 d 點部署的軍力（單位：萬人）：3
請輸入在 e 點部署的軍力（單位：萬人）：4
請輸入在 f 點部署的軍力（單位：萬人）：5
請輸入在 g 點部署的軍力（單位：萬人）：5
請輸入在 h 點部署的軍力（單位：萬人）：4
請輸入在 i 點部署的軍力（單位：萬人）：5
請輸入在 j 點部署的軍力（單位：萬人）：8
請輸入在 k 點部署的軍力（單位：萬人）：8
請輸入在 l 點部署的軍力（單位：萬人）：5
請輸入在 m 點部署的軍力（單位：萬人）：6
請輸入在 n 點部署的軍力（單位：萬人）：7
請輸入在 o 點部署的軍力（單位：萬人）：7
請輸入在 p 點部署的軍力（單位：萬人）：6
恭喜指揮官勝利！
*   *   *   ***   *   *
*   *   *   *   *   *   *
*   *   *   *   *   *   *
*   *   *   ***   *   *

**       **       **
***      ***      ***
*****  *****  *****
*****  *****  *****
*****  *****  *****
*****  *****  *****

```

圖五、自訂測資 1

共調動 $3+5+2+5+6+5+6+7+8+10+9+8+4+8+9+9=105$ 萬人之軍力，因超過可供調動之軍力，故需重新輸入分配軍力。

我方受到敵軍進攻，請指揮官下達命令，部署軍力！
a b c b
e f g h
i j k l
m n o p
我方共有80萬人的軍力可供指揮官調動！
請輸入在 a 點部署的軍力（單位：萬人）：4
請輸入在 b 點部署的軍力（單位：萬人）：5
請輸入在 c 點部署的軍力（單位：萬人）：2
請輸入在 d 點部署的軍力（單位：萬人）：5
請輸入在 e 點部署的軍力（單位：萬人）：6
請輸入在 f 點部署的軍力（單位：萬人）：5
請輸入在 g 點部署的軍力（單位：萬人）：6
請輸入在 h 點部署的軍力（單位：萬人）：7
請輸入在 i 點部署的軍力（單位：萬人）：8
請輸入在 j 點部署的軍力（單位：萬人）：10
請輸入在 k 點部署的軍力（單位：萬人）：9
請輸入在 l 點部署的軍力（單位：萬人）：8
請輸入在 m 點部署的軍力（單位：萬人）：4
請輸入在 n 點部署的軍力（單位：萬人）：8
請輸入在 o 點部署的軍力（單位：萬人）：9
請輸入在 p 點部署的軍力（單位：萬人）：9
指揮官調動的軍力超出可供調動的軍力，請指揮官重新考慮！
a b c b
e f g h
i j k l
m n o p
我方共有80萬人的軍力可供指揮官調動！
請輸入在 a 點部署的軍力（單位：萬人）：1
請輸入在 b 點部署的軍力（單位：萬人）：2
請輸入在 c 點部署的軍力（單位：萬人）：2
請輸入在 d 點部署的軍力（單位：萬人）：1
請輸入在 e 點部署的軍力（單位：萬人）：1
請輸入在 f 點部署的軍力（單位：萬人）：3
請輸入在 g 點部署的軍力（單位：萬人）：3
請輸入在 h 點部署的軍力（單位：萬人）：4
請輸入在 i 點部署的軍力（單位：萬人）：3
請輸入在 j 點部署的軍力（單位：萬人）：5
請輸入在 k 點部署的軍力（單位：萬人）：5
請輸入在 l 點部署的軍力（單位：萬人）：4

圖六、自訂測資 2.1

重新分配之軍力為 $1+2+2+1+1+3+3+4+3+5+5+4+4+3+4+3=48$ 萬人之軍力，並未超過可供調動之軍力，故不用重新輸入分配軍力。但因在 12 個點的軍力都不足，故我軍失利。因此時光機啟動，需重新輸入分配軍力。又因重新輸入後，在 h、j、k、m、n 點位失利，故仍是我軍勝利。

```

請輸入在 m 點部署的軍力 (單位：萬人) : 4
請輸入在 n 點部署的軍力 (單位：萬人) : 3
請輸入在 o 點部署的軍力 (單位：萬人) : 4
請輸入在 p 點部署的軍力 (單位：萬人) : 3
敵軍陣列比我方強大，我方失利
在 a d e f g i j k m n o p 點我方軍力不足！
我方此次共損失19萬人
我方此次共倖存3萬人
為獲取勝利，我方將開啟時光機使時間回到戰爭開始之前，請指揮官重新思考並調動軍力！
時光機已開啟！
a b c b
e f g h
i j k l
m n o p
我方共有80萬人的軍力可供指揮官調動！
請輸入在 a 點部署的軍力 (單位：萬人) : 3
請輸入在 b 點部署的軍力 (單位：萬人) : 2
請輸入在 c 點部署的軍力 (單位：萬人) : 2
請輸入在 d 點部署的軍力 (單位：萬人) : 3
請輸入在 e 點部署的軍力 (單位：萬人) : 4
請輸入在 f 點部署的軍力 (單位：萬人) : 4
請輸入在 g 點部署的軍力 (單位：萬人) : 4
請輸入在 h 點部署的軍力 (單位：萬人) : 2
請輸入在 i 點部署的軍力 (單位：萬人) : 5
請輸入在 j 點部署的軍力 (單位：萬人) : 6
請輸入在 k 點部署的軍力 (單位：萬人) : 6
請輸入在 l 點部署的軍力 (單位：萬人) : 5
請輸入在 m 點部署的軍力 (單位：萬人) : 5
請輸入在 n 點部署的軍力 (單位：萬人) : 5
請輸入在 o 點部署的軍力 (單位：萬人) : 5
請輸入在 p 點部署的軍力 (單位：萬人) : 6
恭喜指揮官勝利！
*   *   *   ***   *   *
*   *   *   *   * * *
* * * *   *   * * *
*   *   ***   *   *

**           **           **
***         ***         ***

```

圖六、自訂測資 2.2