

韌體實驗第12題

Ting-Yu Lin, Adjunct Lecturer (林庭宇)

Office: AMOS Lab E517

Email: tonylin0413@gmail.com



AMOS Lab.
Advanced Mixed-Operation System Lab.
Dept. of Electrical and Computer Engineering,
Tamkang University, Taiwan



**BIDAS Technology
GrounTruth Workshop**
Computer Vision
Image Processing
Edge Computing

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

12. 【追蹤試算表中的儲存格/Spreadsheet Tracking】

有一個 r 列 c 欄 ($1 \leq r, c \leq 50$) 的試算表，列從上到下編號為 $1 \sim r$ ，欄從左到右編號為 $1 \sim c$ 。如圖 (a) 所示，如果先刪除第 1、5 列，然後刪除第 3、6、7、9 欄，結果如圖 (b) 和 (c) 所示。

↖	1	2	3	4	5	6	7	8	9
1	22	55	66	77	88	99	10	12	14
2	2	24	6	8	22	12	14	16	18
3	18	19	20	21	22	23	24	25	26
4	24	25	26	67	22	69	70	71	77
5	68	78	79	80	22	25	28	29	30
6	16	12	11	10	22	56	57	58	59
7	33	34	35	36	22	38	39	40	41

(a)

↖	1	2	3	4	5	6	7	8	9
1	2	24	6	8	22	12	14	16	18
2	18	19	20	21	22	23	24	25	26
3	24	25	26	67	22	69	70	71	77
4	16	12	11	10	22	56	57	58	59
5	33	34	35	36	22	38	39	40	41

(b)

↖	1	2	3	4	5
1	2	24	8	22	16
2	18	19	21	22	25
3	24	25	67	22	71
4	16	12	10	22	58
5	33	34	36	22	40

(c)

第十二題

↘	1	2	3	4	5
1	2	24	8	22	16
2	18	19	21	22	25
3	24	25	67	22	71
4	16	12	10	22	58
5	33	34	36	22	40

(c)

接下來在第 2、3、5 **列前** 各插入一個空列，然後在第 3 **欄前** 插入一個空欄，會得到如圖 (d) 和 (e) 的結果。

↘	1	2	3	4	5
1	2	24	8	22	16
2					
3	18	19	21	22	25
4					
5	24	25	67	22	71
6	16	12	10	22	58
7					
8	33	34	36	22	40

(d)

↘	1	2	3	4	5	6
1	2	24		8	22	16
2						
3	18	19		21	22	25
4						
5	24	25		67	22	71
6	16	12		10	22	58
7						
8	33	34		36	22	40

(e)

第十二題

淡江大學電機工程學系 106 學年度韌體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX r1 c1 r2 c2 交換儲存格 (r1, c1), (r2, c2)
- <command> A x1 x2... xA 插入或刪除 A 列或欄 (DC-刪除欄, DR-刪除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/刪除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「r c」，表示查詢原始表格的儲存格 (r, c)。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1):

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2):

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

➤ 輸入列 $r = 7$ ，欄 $c = 9$ 。

輸入 (註 1)：

7 9

5

DR 2 1 5

DC 4 3 6 7 9

IC 1 3

IR 2 2 4

EX 1 2 6 5

4

4 8

5 5

7 8

6 5

0 0

輸出 (註 2)：

Spreadsheet #1

Cell data in (4, 8) moved to (4, 6)

Cell data in (5, 5) GONE

Cell data in (7, 8) moved to (7, 6)

Cell data in (6, 5) moved to (1, 2)

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

➤ 輸入列 $r = 7$ ，欄 $c = 9$ 。

➤ 輸入操作數量 $n = 5$ 。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX r1 c1 r2 c2 交換儲存格 $(r1, c1)$, $(r2, c2)$
- <command> A x1 x2... xA 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「r c」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1):

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2):

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1):

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2):

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX r_1 c_1 r_2 c_2 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A x_1 x_2 ... x_A 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 r c 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1):

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2):

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。

第十二題

淡江大學電機工程學系 106 學年度韌體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。
- 4 個查詢。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX r1 c1 r2 c2 交換儲存格 (r1, c1), (r2, c2)
- <command> A x1 x2... xA 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「r c」，表示查詢原始表格的儲存格 (r, c)。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1):

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2):

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。
- 4 個查詢。
- (4, 8) 經過五次操作後移動到 (4, 6)。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX r1 c1 r2 c2 交換儲存格 (r1, c1), (r2, c2)
- <command> A x1 x2... xA 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「r c」，表示查詢原始表格的儲存格 (r, c)。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。
- 4 個查詢。
- (4, 8) 經過五次操作後移動到 (4, 6)。
- (5, 5) 被删除了。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1):

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2):

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。
- 4 個查詢。
- (4, 8) 經過五次操作後移動到 (4, 6)。
- (5, 5) 被删除了。
- (7, 8) 經過五次操作後移動到 (7, 6)。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。
- 4 個查詢。
- (4, 8) 經過五次操作後移動到 (4, 6)。
- (5, 5) 被删除了。
- (7, 8) 經過五次操作後移動到 (7, 6)。
- (6, 5) 經過五次操作後移動到 (1, 2)。

第十二題

淡江大學電機工程學系 106 學年度軟體實驗題目

你的任務是模擬這樣的 n 個操作，具體來說一共有 5 種操作：

- EX $r_1\ c_1\ r_2\ c_2$ 交換儲存格 (r_1, c_1) , (r_2, c_2)
- <command> A $x_1\ x_2\ \dots\ x_A$ 插入或删除 A 列或欄 (DC-删除欄, DR-删除列, IC-插入欄, IR-插入列, $1 \leq A \leq 10$)

在插入/删除指令後，各個 x 值不同，且順序任意。接下來是 q 個查詢，每個查詢格式為「 $r\ c$ 」，表示查詢原始表格的儲存格 (r, c) 。對於每個查詢，輸出操作執行完成後該儲存格的新位置。輸入保證在任何時刻列欄數均不超過 50。

輸入 (註 1)：

```
7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0
```

輸出 (註 2)：

```
Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)
```

- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR 删除 2 列，第 1 和 5 列。
- DC 删除 4 欄，第 3, 6, 7 和 9 欄。
- IC 插入 1 欄，第 3 欄。
- IR 插入 2 列，第 2 和 4 列。
- EX(1, 2) 與 (6, 5) 交換儲存格。
- 4 個查詢。
- (4, 8) 經過五次操作後移動到 (4, 6)。
- (5, 5) 被删除了。
- (7, 8) 經過五次操作後移動到 (7, 6)。
- (6, 5) 經過五次操作後移動到 (1, 2)。
- (0, 0) 代表結束。

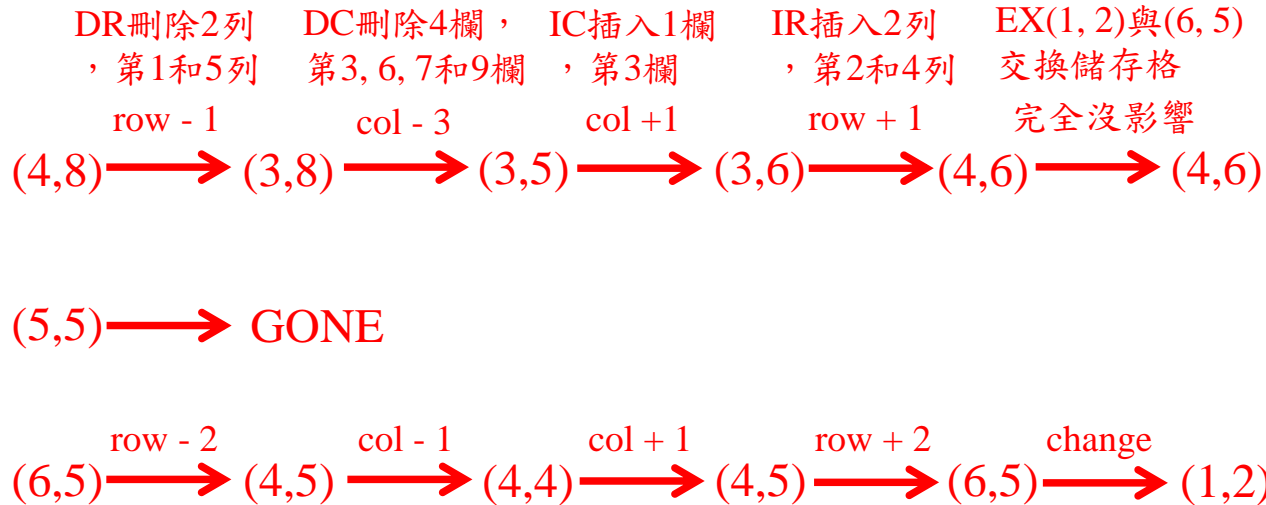
第十二題

輸入 (註 1):

7 9
5
DR 2 1 5
DC 4 3 6 7 9
IC 1 3
IR 2 2 4
EX 1 2 6 5
4
4 8
5 5
7 8
6 5
0 0

輸出 (註 2):

Spreadsheet #1
Cell data in (4, 8) moved to (4, 6)
Cell data in (5, 5) GONE
Cell data in (7, 8) moved to (7, 6)
Cell data in (6, 5) moved to (1, 2)



- 輸入列 $r = 7$ ，欄 $c = 9$ 。
- 輸入操作數量 $n = 5$ 。
- DR刪除2列，第1和5列。
- DC刪除4欄，第3, 6, 7和9欄。
- IC插入1欄，第3欄。
- IR插入2列，第2和4列。
- EX(1, 2)與(6, 5)交換儲存格。
- 4個查詢。
- (4, 8)經過五次操作後移動到(4, 6)。
- (5, 5)被刪除了。
- (7, 8)經過五次操作後移動到(7, 6)。
- (6, 5)經過五次操作後移動到(1, 2)。
- (0, 0)代表結束。

第十二題

➤ 題目解析：

- (1) 先將所有操作儲存，然後對每個查詢重新執行這些操作。
- (2) 使用 `vector<type>` 向量儲存任意長度的 `type` 型態陣列。

第十二題

```

1  #define _CRT_SECURE_NO_WARNINGS
2  //[追蹤試算表中的儲存格/Spreadsheet Tracking](1/3)
3  #define IN "P12IN.txt"
4  #define OUT "P12OUT.txt"
5  //*****
6  #include <iostream>
7  #include <ctime>
8  using namespace std;
9  void redir(void);
10
11 //*****
12 /* Work Space*/
13 #include <vector>
14 typedef struct command {
15     char c[3]; //"EX", "DC", "DR", "IC", "IR"
16     int r1, c1, r2, c2;
17     int a;
18     int x[10]; //依題意
19 }CMD;
20 vector<CMD> cmd;
21
22 int r, c, n;
23 int simulate(int *r0, int *c0);
24 //*****
25
26 int main(void) {
27     redir(); //redirection
28

```

需要修改或自行撰寫的部分

- 初行加入crt secure no warnings。
- 記得修改輸入和輸出txt檔案名稱。
- #define為C++直接定義，變數型態會依照定義自動判定。
- #include <vector>內建函式。
- 宣告struct型態、名為command的結構，並令此結構的別名為CMD。
- 宣告名為cmd、型態為CMD結構的vector(向量)。
- 宣告整數全域變數r：列、c：欄、n：操作次數。
- 宣告simulate執行操作副函式標頭檔。

```

66     freopen("CON", "r", stdin); //取消重新導向
67     freopen("CON", "w", stdout);
68     printf("Time used = %.2f\n", (double)clock() / CLK_TCK);
69     system("pause");
70     return 0; //the end...
71 }
72
73 void redir(void) {
74     freopen(IN, "r", stdin);
75     freopen(OUT, "w", stdout);
76 }

```


第十二題

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈讀取輸入儲存格資訊

(3) 儲存操作

(4) 顯示查詢結果，其中使用執行操作副函式

```

29 //*****
30 /* Work Space*/
31 int i, j, kase = 0, q, r0, c0;
32
33 while (scanf("%d%d%d", &r, &c, &n) == 3) {
34     for (i = 0; i < n; i++) {
35         cmd.resize(cmd.size() + 1);
36         scanf("%s", cmd[i].c);
37         if (cmd[i].c[0] == 'E') {
38             scanf("%d%d%d%d", &cmd[i].r1, &cmd[i].c1, &cmd[i].r2, &cmd[i].c2);
39         }
40         else {
41             scanf("%d", &cmd[i].a);
42             for (j = 0; j < cmd[i].a; j++) {
43                 scanf("%d", &cmd[i].x[j]);
44             }
45         }
46     }
47     if (kase > 0) {
48         printf("\n");
49     }
50     printf("Spreadsheet #%d\n", ++kase);
51     scanf("%d", &q);
52     while (q--) {
53         scanf("%d%d", &r0, &c0);
54         printf("Cell data in (%d, %d) ", r0, c0);
55         if (!simulate(&r0, &c0)) {
56             printf("GONE\n");
57         }
58         else {
59             printf("moved to (%d, %d)\n", r0, c0);
60         }
61     }
62 }
63 //*****

```

第十二題

輸入 (註 1): 輸出 (註 2):
 7 9 Spreadsheet #1
 5 Cell data in (4, 8) moved to (4, 6)
 DR 2 1 5 Cell data in (5, 5) GONE
 DC 4 3 6 7 9 Cell data in (7, 8) moved to (7, 6)
 IC 1 3 Cell data in (6, 5) moved to (1, 2)
 IR 2 2 4
 EX 1 2 6 5
 4
 4 8
 5 5
 7 8
 6 5
 0 0

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈讀取輸入儲存格資訊

(3) 儲存操作

(4) 顯示查詢結果，其中使用執行操作副函式

● Line 31:

✓ 宣告**整數**變數**kase**：第幾次儲存格操作set。

✓ 宣告**整數**變數**q**：查詢的次數。

✓ 宣告**整數**變數**r0**：要查詢的列。

✓ 宣告**整數**變數**c0**：要查詢的欄。

● Line 33:

✓ scanf**列、欄、操作次數**至變數**r, c, n**，如果三個變數皆成功輸入則回傳3，while迴圈執行。

```

29 //*****
30 /* Work Space*/
31 int i, j, kase = 0, q, r0, c0;
32
33 while (scanf("%d%d%d", &r, &c, &n) == 3) {
34     for (i = 0; i < n; i++) {
35         cmd.resize(cmd.size() + 1);
36         scanf("%s", cmd[i].c);
37         if (cmd[i].c[0] == 'E') {
38             scanf("%d%d%d%d", &cmd[i].r1, &cmd[i].c1, &cmd[i].r2, &cmd[i].c2);
39         }
40         else {
41             scanf("%d", &cmd[i].a);
42             for (j = 0; j < cmd[i].a; j++) {
43                 scanf("%d", &cmd[i].x[j]);
44             }
45         }
46     }
47     if (kase > 0) {
48         printf("\n");
49     }
50     printf("Spreadsheet #%d\n", ++kase);
51     scanf("%d", &q);
52     while (q--) {
53         scanf("%d%d", &r0, &c0);
54         printf("Cell data in (%d, %d) ", r0, c0);
55         if (!simulate(&r0, &c0)) {
56             printf("GONE\n");
57         }
58         else {
59             printf("moved to ");
60         }
61     }
62 }
63 //*****

```

```

13 #include <vector>
14 typedef struct command {
15     char c[3]; // "EX", "DC", "DR", "IC", "IR"
16     int r1, c1, r2, c2;
17     int a;
18     int x[10]; //依題意
19 }CMD;
20 vector<CMD> cmd;
21
22 int r, c, n;
23 int simulate(int *r0, int *c0);

```

第十二題

輸入 (註 1): 輸出 (註 2):

7 9	Spreadsheet #1
5	Cell data in (4, 8) moved to (4, 6)
DR 2 1 5	Cell data in (5, 5) GONE
DC 4 3 6 7 9	Cell data in (7, 8) moved to (7, 6)
IC 1 3	Cell data in (6, 5) moved to (1, 2)
IR 2 2 4	
EX 1 2 6 5	
4	
4 8	
5 5	
7 8	
6 5	
0 0	

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈讀取輸入儲存格資訊

(3) 儲存操作

(4) 顯示查詢結果，其中使用執行操作副函式

● Line 34-35:

✓ for迴圈儲存操作，範圍為操作次數(0 ~ n-1)。

✓ cmd.resize：重新調整vector的大小。

✓ cmd.size() + 1：將vector的大小加一。

Member Functions

• size_type **vector::size()**;

- Return the **number of elements in the vector**
- size_type is an alias of **unsigned int**

- 回傳vector當中元素的數量。
- size_type是unsigned int的別名。

• void **vector::resize**(size_type n);

- **Resize** the **vector** so that it contains *n* elements
- If *n* is greater than the current size, the vector is expanded

- 調整vector的大小到*n*個元素。
- 如果*n*大於當前的尺寸，則vector會擴增

```

29  //*****
30  /* Work Space*/
31  int i, j, kase = 0, q, r0, c0;
32
33  while (scanf("%d%d%d", &r, &c, &n) == 3) {
34      for (i = 0; i < n; i++) {
35          cmd.resize(cmd.size() + 1);
36          scanf("%s", cmd[i].c);
37          if (cmd[i].c[0] == 'E') {
38              scanf("%d%d%d%d", &cmd[i].r1, &cmd[i].c1, &cmd[i].r2, &cmd[i].c2);
39          }
40          else {
41              scanf("%d", &cmd[i].a);
42              for (j = 0; j < cmd[i].a; j++) {
43                  scanf("%d", &cmd[i].x[j]);
44              }
45          }
46      }
47      if (kase > 0) {
48          printf("\n");
49      }
50      printf("Spreadsheet #%d\n", ++kase);
51      scanf("%d", &q);
52      while (q--) {
53          scanf("%d%d", &r0, &c0);
54          printf("Cell data in (%d, %d) ", r0, c0);
55          if (!simulate(&r0, &c0)) {
56              printf("GONE\n");
57          }
58          else {
59              printf("moved to ");
60          }
61      }
62  }
63  //*****

```

```

13  #include <vector>
14  typedef struct command {
15      char c[3]; // "EX", "DC", "DR", "IC", "IR"
16      int r1, c1, r2, c2;
17      int a;
18      int x[10]; //依題意
19  } CMD;
20  vector<CMD> cmd;
21
22  int r, c, n;
23  int simulate(int *r0, int *c0);

```

第十二題

輸入 (註1): 輸出 (註2):
 7 9 Spreadsheet #1
 5 Cell data in (4, 8) moved to (4, 6)
 DR 2 1 5 Cell data in (5, 5) GONE
 DC 4 3 6 7 9 Cell data in (7, 8) moved to (7, 6)
 IC 1 3 Cell data in (6, 5) moved to (1, 2)
 IR 2 2 4
 EX 1 2 6 5
 4
 4 8
 5 5
 7 8
 6 5
 0 0

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈讀取輸入儲存格資訊

(3) 儲存操作

(4) 顯示查詢結果，其中使用執行操作副函式

● Line 36:

✓ scanf指令字串(DR/IR/DC/IC/EX)至vector結構cmd的字元陣列c。

● Line 37-39:

✓ if判斷指令是否為EX交換儲存格，如果是則執行Line 38依序scanf兩個儲存格的列位和欄位至vector結構cmd的r1, c1, r2, c2。

✓ Line 40-45:

✓ else否的話指令可能為DC、DR、IC、IR操作，scanf要刪除或增加的數量至vector結構cmd的a，for迴圈依序讀入要刪除或增加的列位或欄位至vector結構cmd的x[j]。

```

29 //*****
30 /* Work Space*/
31 int i, j, kase = 0, q, r0, c0;
32
33 while (scanf("%d%d%d", &r, &c, &n) == 3) {
34     for (i = 0; i < n; i++) {
35         cmd.resize(cmd.size() + 1);
36         scanf("%s", cmd[i].c);
37         if (cmd[i].c[0] == 'E') {
38             scanf("%d%d%d%d", &cmd[i].r1, &cmd[i].c1, &cmd[i].r2, &cmd[i].c2);
39         }
40         else {
41             scanf("%d", &cmd[i].a);
42             for (j = 0; j < cmd[i].a; j++) {
43                 scanf("%d", &cmd[i].x[j]);
44             }
45         }
46     }
47     if (kase > 0) {
48         printf("\n");
49     }
50     printf("Spreadsheet #%d\n", ++kase);
51     scanf("%d", &q);
52     while (q--) {
53         scanf("%d%d", &r0, &c0);
54         printf("Cell data in (%d, %d) ", r0, c0);
55         if (!simulate(&r0, &c0)) {
56             printf("GONE\n");
57         }
58         else {
59             printf("moved to ");
60         }
61     }
62 }
63 //*****

```

```

13 #include <vector>
14 typedef struct command {
15     char c[3]; // "EX", "DC", "DR", "IC", "IR"
16     int r1, c1, r2, c2;
17     int a;
18     int x[10]; // 依題意
19 } CMD;
20 vector<CMD> cmd;
21
22 int r, c, n;
23 int simulate(int *r0, int *c0);

```


第十二題

輸入 (註 1): 輸出 (註 2):
 7 9 Spreadsheet #1
 5 Cell data in (4, 8) moved to (4, 6)
 DR 2 1 5 Cell data in (5, 5) GONE
 DC 4 3 6 7 9 Cell data in (7, 8) moved to (7, 6)
 IC 1 3 Cell data in (6, 5) moved to (1, 2)
 IR 2 2 4
 EX 1 2 6 5
 4
 4 8
 5 5
 7 8
 6 5
 0 0

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈讀取輸入儲存格資訊

(3) 儲存操作

(4) 顯示查詢結果，其中使用執行操作副函式

● Line 47-50:

✓ 判斷是否為**第一回合**，如果**kase**大於0表示**至少執行過一回合**，則顯示換行，然後顯示字串Spreadsheet...

● Line 51-53:

✓ scanf**查詢次數**至變數q。

✓ while(q--)依序查詢**要查詢的儲存格(r0, c0)**。

● Line 54+59:

✓ 如果要查詢的儲存格(r0, c0)**存在**，則**顯示移動的資訊**。

● Line 55-57:

✓ 如果要查詢的儲存格(r0, c0)**被刪除**，則顯示**GONE**。副函式simulate執行操作，回傳為0則代表要查詢的(r0, c0)**被刪除**，回傳為1則代表**移動成功**。

```

29 //*****
30 /* Work Space*/
31 int i, j, kase = 0, q, r0, c0;
32
33 while (scanf("%d%d%d", &r, &c, &n) == 3) {
34     for (i = 0; i < n; i++) {
35         cmd.resize(cmd.size() + 1);
36         scanf("%s", cmd[i].c);
37         if (cmd[i].c[0] == 'E') {
38             scanf("%d%d%d%d", &cmd[i].r1, &cmd[i].c1, &cmd[i].r2, &cmd[i].c2);
39         }
40         else {
41             scanf("%d", &cmd[i].a);
42             for (j = 0; j < cmd[i].a; j++) {
43                 scanf("%d", &cmd[i].x[j]);
44             }
45         }
46     }
47     if (kase > 0) {
48         printf("\n");
49     }
50     printf("Spreadsheet #%d\n", ++kase);
51     scanf("%d", &q);
52     while (q--) {
53         scanf("%d%d", &r0, &c0);
54         printf("Cell data in (%d, %d) ", r0, c0);
55         if (!simulate(&r0, &c0)) {
56             printf("GONE\n");
57         }
58         else {
59             printf("moved to (%d, %d)\n", r0, c0);
60         }
61     }
62 }
63 //*****

```

第十二題

```

80 int simulate(int *r0, int *c0) {
81     int x, i, j, dr, dc; //列變化量, 行變化量
82     for (i = 0; i < n; i++) {
83         if (cmd[i].c[0] == 'E') { //"EX"
84             if (cmd[i].r1 == *r0 && cmd[i].c1 == *c0) {
85                 *r0 = cmd[i].r2;
86                 *c0 = cmd[i].c2;
87             }
88             else if (cmd[i].r2 == *r0 && cmd[i].c2 == *c0) {
89                 *r0 = cmd[i].r1;
90                 *c0 = cmd[i].c1;
91             }
92         }
93         else { //計算列變化量, 行變化量
94             dr = dc = 0;
95             for (j = 0; j < cmd[i].a; j++) {
96                 x = cmd[i].x[j];
97                 if (cmd[i].c[0] == 'I') {
98                     if (cmd[i].c[1] == 'R' && *r0 >= x) dr++; //"IR"
99                     if (cmd[i].c[1] == 'C' && *c0 >= x) dc++; //"IC"
100                 }
101                 else { //cmd[i].c[0] == 'D'
102                     if (cmd[i].c[1] == 'R' && *r0 == x) return 0; //"DR" : 所在列被刪除
103                     if (cmd[i].c[1] == 'C' && *c0 == x) return 0; //"DC" : 所在欄被刪除
104                     if (cmd[i].c[1] == 'R' && *r0 > x) dr--; //"DR"
105                     if (cmd[i].c[1] == 'C' && *c0 > x) dc--; //"DC"
106                 }
107             }
108             *r0 += dr; //更新
109             *c0 += dc;
110         }
111     }
112     return 1;
113 }

```

```

13 #include <vector>
14 typedef struct command {
15     char c[3]; //"EX", "DC", "DR", "IC", "IR"
16     int r1, c1, r2, c2;
17     int a;
18     int x[10]; //依題意
19 }CMD;
20 vector<CMD> cmd;
21
22 int r, c, n;
23 int simulate(int *r0, int *c0);

```

需要修改或自行撰寫的部分

```

55 if (!simulate(&r0, &c0)) {
56     printf("GONE\n");
57 }

```

- simulate副函式：(r0, c0)為要查詢的儲存格。參數設定為**指標**型態目的為在副函式中改變r0與c0的值，在Main主函式中對應的參數也會跟著改變。
- Line 81:
 - ✓ 宣告**整數**變數**dr**列變化量，**dc**行變化量。
- Line 83-92:
 - ✓ 如果操作型態為**EX**交換儲存格，當r1 = r0且c1 = c0則直接調換結果為(r2, c2)，**else if**反之。
- Line 93-111:
 - ✓ 其餘操作型態，先**初始化**列和欄變化量為0，**for**迴圈依序操作，**cmd[i].a**為要刪除或增加的**數量**，**cmd[i].x[j]**為要刪除或增加的**列位或欄位**。
- Line 97-99:
 - ✓ 如果操作型態為**插入**，且要查詢的列或欄**大於**要插入的位置，則其**變化量加一**。
- Line 102-103:
 - ✓ 如果操作型態為**刪除**，且要查詢的列或欄**等於**要刪除的位置，則**return 0**，表示儲存格做刪除的動作。

第十二題

```

80 int simulate(int *r0, int *c0) {
81     int x, i, j, dr, dc; //列變化量, 行變化量
82     for (i = 0; i < n; i++) {
83         if (cmd[i].c[0] == 'E') { //"EX"
84             if (cmd[i].r1 == *r0 && cmd[i].c1 == *c0) {
85                 *r0 = cmd[i].r2;
86                 *c0 = cmd[i].c2;
87             }
88             else if (cmd[i].r2 == *r0 && cmd[i].c2 == *c0) {
89                 *r0 = cmd[i].r1;
90                 *c0 = cmd[i].c1;
91             }
92         }
93         else { //計算列變化量, 行變化量
94             dr = dc = 0;
95             for (j = 0; j < cmd[i].a; j++) {
96                 x = cmd[i].x[j];
97                 if (cmd[i].c[0] == 'I') {
98                     if (cmd[i].c[1] == 'R' && *r0 >= x) dr++; //"IR"
99                     if (cmd[i].c[1] == 'C' && *c0 >= x) dc++; //"IC"
100                 }
101                 else { //cmd[i].c[0] == 'D'
102                     if (cmd[i].c[1] == 'R' && *r0 == x) return 0; //"DR" : 所在列被刪除
103                     if (cmd[i].c[1] == 'C' && *c0 == x) return 0; //"DC" : 所在欄被刪除
104                     if (cmd[i].c[1] == 'R' && *r0 > x) dr--; //"DR"
105                     if (cmd[i].c[1] == 'C' && *c0 > x) dc--; //"DC"
106                 }
107             }
108             *r0 += dr; //更新
109             *c0 += dc;
110         }
111     }
112     return 1;
113 }

```

```

13 #include <vector>
14 typedef struct command {
15     char c[3]; //"EX", "DC", "DR", "IC", "IR"
16     int r1, c1, r2, c2;
17     int a;
18     int x[10]; //依題意
19 }CMD;
20 vector<CMD> cmd;
21
22 int r, c, n;
23 int simulate(int *r0, int *c0);

```

需要修改或自行撰寫的部分

```

55 if (!simulate(&r0, &c0)) {
56     printf("GONE\n");
57 }

```

- simulate副函式：(r0, c0)為要查詢的儲存格。參數設定為**指標**型態目的為在副函式中改變r0與c0的值，在Main主函式中對應的參數也會跟著改變。
- Line 104-105:
 - ✓ 如果操作型態為**刪除**，且要查詢的列或欄**大於**要插入的位置，則其**變化量減一**。
- Line 108-109:
 - ✓ 更新要查詢的列和欄位。
- Line 112:
 - ✓ 回傳1則為正常移動。

第十二題

P12IN.txt - 記事本	P12OUT.txt - 記事本
檔案(F) 編輯(E) 格式	檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
7 9	Spreadsheet #1
5	Cell data in (4, 8) moved to (4, 6)
DR 2 1 5	Cell data in (5, 5) GONE
DC 4 3 6 7 9	Cell data in (7, 8) moved to (7, 6)
IC 1 3	Cell data in (6, 5) moved to (1, 2)
IR 2 2 4	
EX 1 2 6 5	
4	
4 8	
5 5	
7 8	
6 5	
0 0	



Q&A

