

Problem 1

- 目標：

從給定的起點探索地圖，並判斷能否到達終點

練習重點：

1. 設計適當的遞迴函式，探索地圖

Problem 1

函式參考 prototype:

```
void explore( int cur_pos[2] );
```

參數列表:

int cur_pos[2] 目前所在位置的座標

Problem 1

函式流程：

```
void explore( int cur_pos[2] );
```

1. 將目前的位置記錄為 “reachable”
2. 從目前所在的位置，向外探索。

嘗試所有可以走的方向 (for loop)：

若該方向可以走，並且還沒走過，

那麼就走過去（遞迴呼叫 `explore` 函式）

Problem 2

- 目標：

計算從給定的起點至終點的路徑

練習重點：

1. 理解遞迴探索方法的流程，並理解如何在這個過程中，紀錄自己需要的資訊。
2. 練習 Back-tracing 技巧，將紀錄起來的 (區域) 資訊片段組合、還原為所需的解答

Problem 2

- 想法：

1. 在遞迴探索的過程裡，對於每個可到達的點，
紀錄『它的前一個點是誰』

```
int fromm[50][50][2]; // 前一個點的座標
```

2. 既然每個可到達的點，都有紀錄來源點。

那麼，從終點往前走，就可以重建出路徑。

函式流程：

```
void explore( int cur_pos[2] );
```

1. 將目前的位置記錄為 “reachable”
2. 從目前所在的位置，向外探索。

嘗試所有可以走的方向 (for loop)：

若該方向可以走，並且還沒走過，

```
/* 遞迴呼叫前，先紀錄來源，再遞迴 */
```

那麼就走過去 (遞迴呼叫 explore 函式)

函式流程：

```
int path[50*50][2], len_of_path;
```

```
void construct_path( int dest[2] );
```

依據 fromm 陣列裡紀錄的資訊，建出路徑

1. 從 dest[2] 開始，持續往 “上一個點” 移動，
直到走到 src[2] 為止。
2. 走的過程，
同時把每個走過的點紀錄在 path 陣列裡
3. 反序印出 path 陣列的內容，即為所求路徑