程式設計與實習(二)

BY 孫茂勛

EMAIL:JOHN85051232@GMAIL.COM

下禮拜（５／２）有小考

- 考Linked List、Stack、Queue的實作
- 怎麼做Linked List的新增、刪除、輸出
- Stack和Queue的相關基本操作

- 這次不能OPEN BOOK (因為答案就在上面了....)

下下禮拜（５／９）有測驗

要進行APCS的C程式設計檢測試作(高中生的程式檢定)
時間:
９:00~10:00預備
10:00~12:00正式測驗
算一次小考成績，並且會有小禮物

Stack

基本操作：

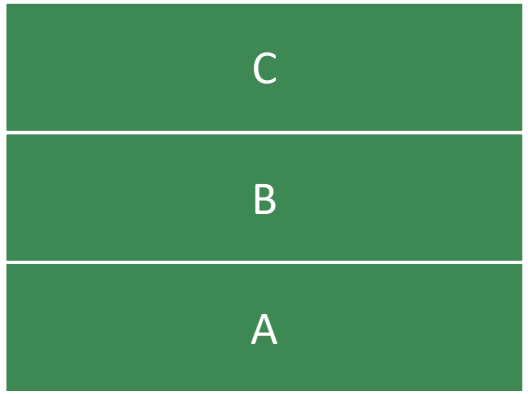
- push
- pop
- top
- size
- empty

Stack

基本操作：

- push

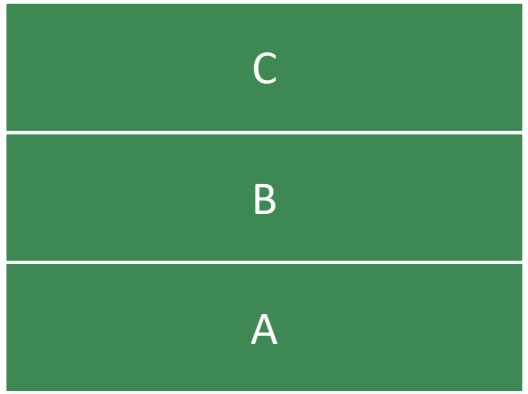
- push A -> push B -> push C



Stack

基本操作：

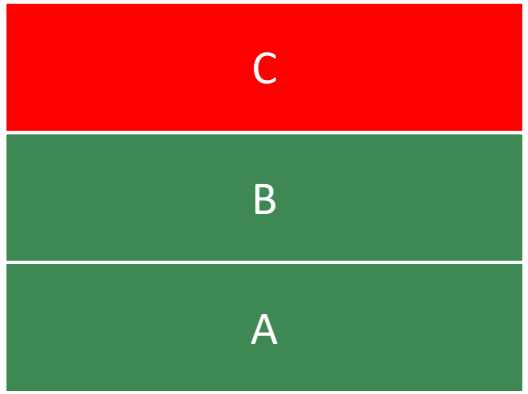
- pop
- pop -> pop -> pop



Stack

基本操作：

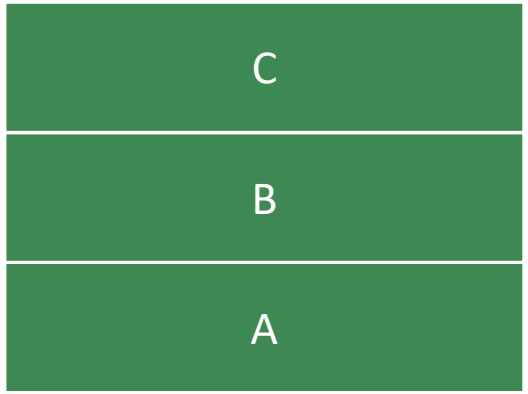
- top



Stack

基本操作：

- size = 3



Stack

基本操作：

- empty

```c
struct stack
{
    int num;
    stack *next;
};
```

```c
int main()
{
    stack *head = new stack;
    head->next = NULL;
    while(1)
    {
        printf("==================================================================\n")
        printf("| Stack相關操作 0:print 1:push 2:pop 3:top 4:size 5:empty  |\n")
        printf("==================================================================\n")
        int n = 0,num = 0;
        scanf("%d",&n);
        switch(n)
        {
            case 0:print(head);break;
            case 1:
                printf("要新增的資料:");
                scanf("%d",&num);
                push(head,num);
                break;
            case 2:pop(head);break;
            case 3:top(head);break;
            case 4:size(head);break;
            case 5:empty(head);break;
        }
    }
    system("pause");
```

print

```
void print(stack *head)
{
    stack *ptr = head;
    printf("目前堆疊裡面的資料有:");
    while(ptr->next != NULL)
    {
        ptr = ptr->next;
        printf("%d ",ptr->num);
    }
    printf("\n");
}
```

push

```
void push(stack *head,int data)
{
    stack *n = new stack;
    n->num = data;
    n->next = NULL;
    stack *ptr = head;
    while(ptr->next != NULL)
    {
        ptr = ptr->next;
    }
    ptr->next = n;
}
```

pop

```
void pop(stack *head)
{
    if(head->next == NULL)
    {
        printf("沒有資料可進行pop()!\n");
    }
    else
    {
        stack *ptr = head;
        while(ptr->next->next != NULL)
        {
            ptr = ptr->next;
        }
        stack *temp = ptr->next;
        delete temp;
        ptr->next = NULL;
    }
}
```

top

```
void top(stack *head)
{
    stack *top = head;
    while(top->next != NULL)
    {
        top = top->next;
    }
    printf("Stack top = %d\n",top->num);
}
```

size

```
void size(stack *head)
{
    int n = 0;
    stack *ptr = head;
    while(ptr->next != NULL)
    {
        ptr = ptr->next;
        n++;
    }
    printf("Stack size = %d\n",n);
}
```

empty

```
void empty(stack *head)
{
    if(head->next == NULL)
    {
        printf("stack目前是空的!\n");
    }
    else
    {
        printf("stack目前不是空的\n");
    }
}
```

Queue

基本操作：

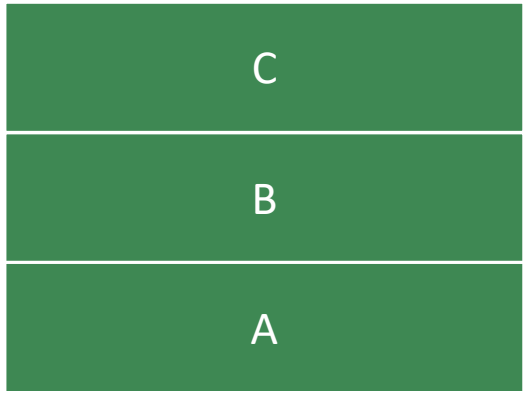
- push
- pop
- front
- size
- empty

Queue

基本操作：

- push

- push A -> push B -> push C

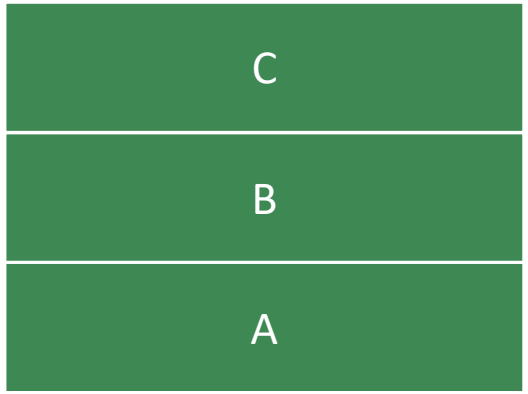


Queue

基本操作：

- pop

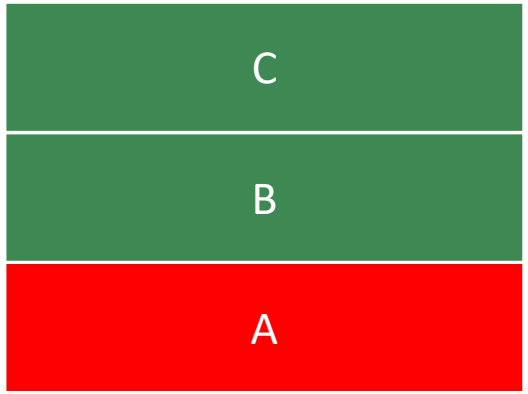
- pop -> pop -> pop



Queue

基本操作：

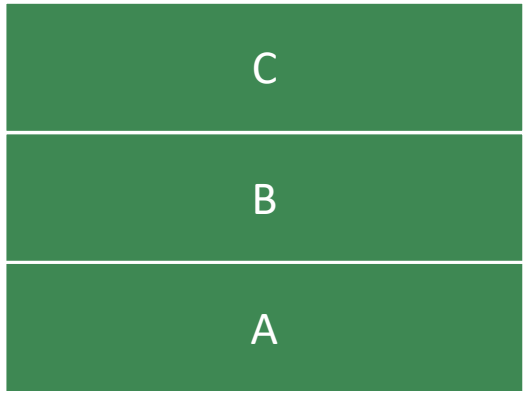
- front



Queue

基本操作：

- size = 3



Queue

基本操作：

- empty

pop

```
queue *pop(queue *head)
{
    if(head->next == NULL)
    {
        printf("沒有資料可進行pop()!\n");
    }
    else
    {
        queue *ptr = head->next;
        delete head;
        return ptr;
    }
}
```

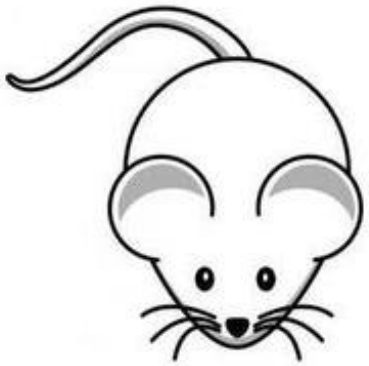
pop

```
{

    printf("==============================================================\n");
    printf("| Queue相關操作 0:print 1:push 2:pop 3:front 4:size 5:empty|\n");
    printf("==============================================================\n");
    int n = 0,num = 0;
    scanf("%d",&n);
    switch(n)
    {
        case 0:print(head);break;
        case 1:
            printf("要新增的資料:");
            scanf("%d",&num);
            push(head,num);
            break;
        case 2:head = pop(head);break;//把新head的位址傳回來
        case 3:front(head);break;
        case 4:size(head);break;
        case 5:empty(head);break;
    }
}
system("pause");
return 0;
}
```

老鼠走迷宮

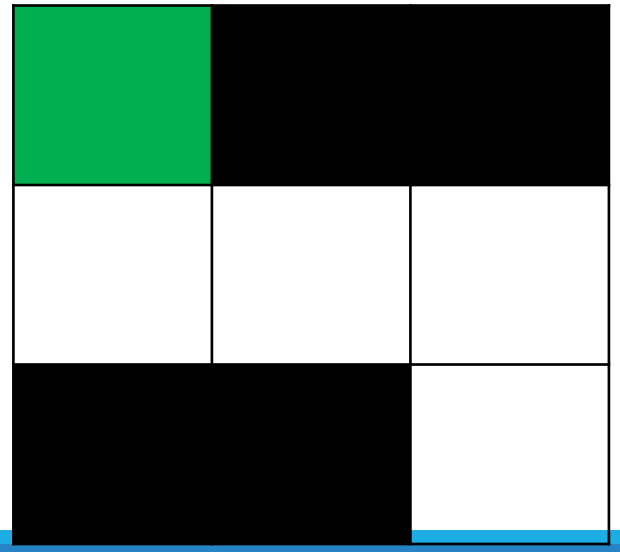
有一個迷宮，在終點（右下角）放置老鼠最愛吃的奶酪，將老鼠放在左上角，老鼠必須穿越迷宮才能找到他的食物，請問老鼠該怎麼走？



老鼠走迷宮

我們怎麼走迷宮的？

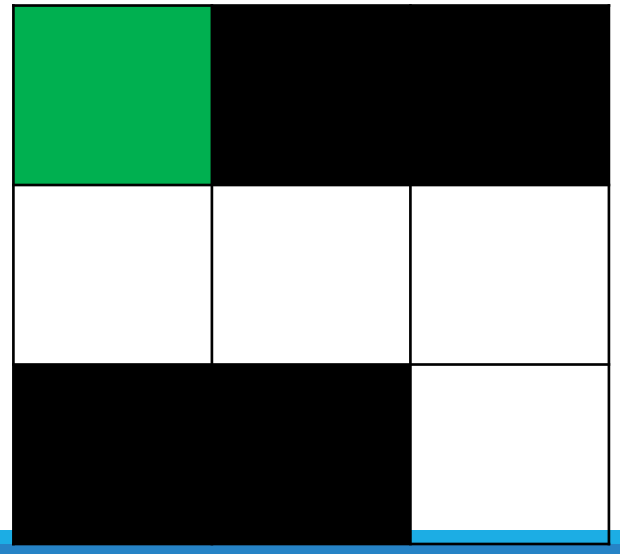
往一個方向走，直到沒路了再走另外一個方向



Q：假設走的先後順序依序是：
右邊->左邊->下面->上面
這張3*3的地圖會怎麼走？

老鼠走迷宮

Q：假設走的先後順序依序是：右邊->左邊->下面->上面
這張3*3的地圖會怎麼走？

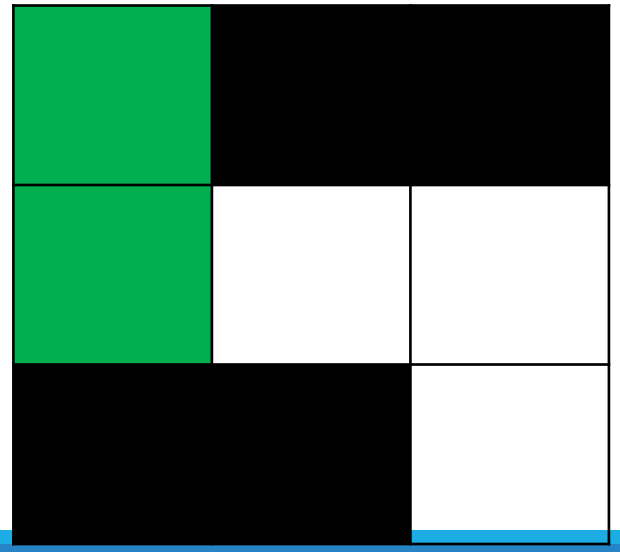


1.向右(X)
2.向左(X)
3.向下(O)

老鼠走迷宮

Q：假設走的先後順序依序是：右邊->左邊->下面->上面
這張3*3的地圖會怎麼走？

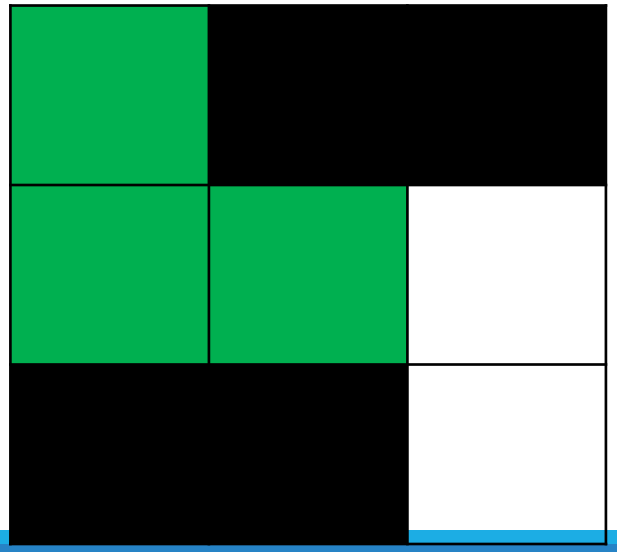
1.向右(O)



老鼠走迷宮

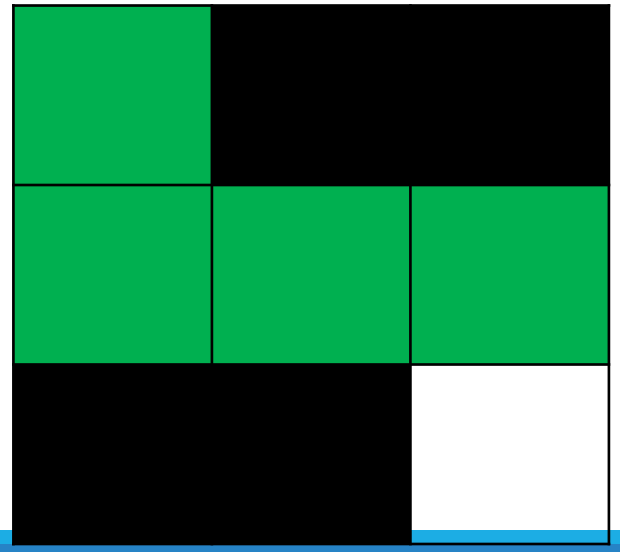
Q：假設走的先後順序依序是：右邊->左邊->下面->上面
這張3*3的地圖會怎麼走？

1.向右(O)



老鼠走迷宮

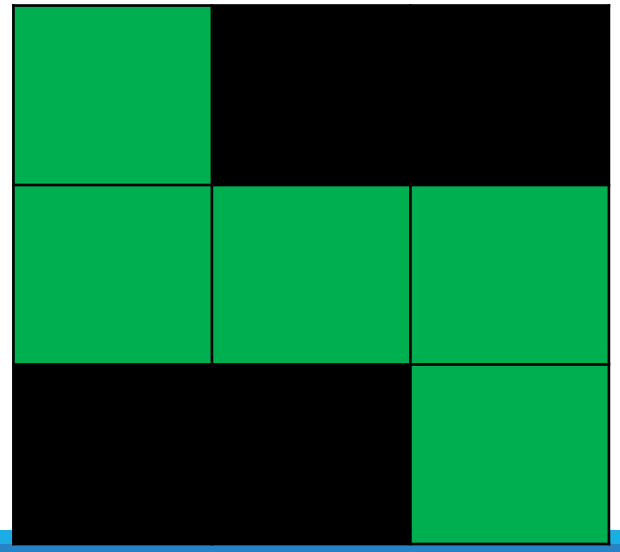
Q：假設走的先後順序依序是：右邊->左邊->下面->上面
這張3*3的地圖會怎麼走？



1.向右(X)
2.向左(X) ->走過的就不能再走了，why?
3.向下(O)

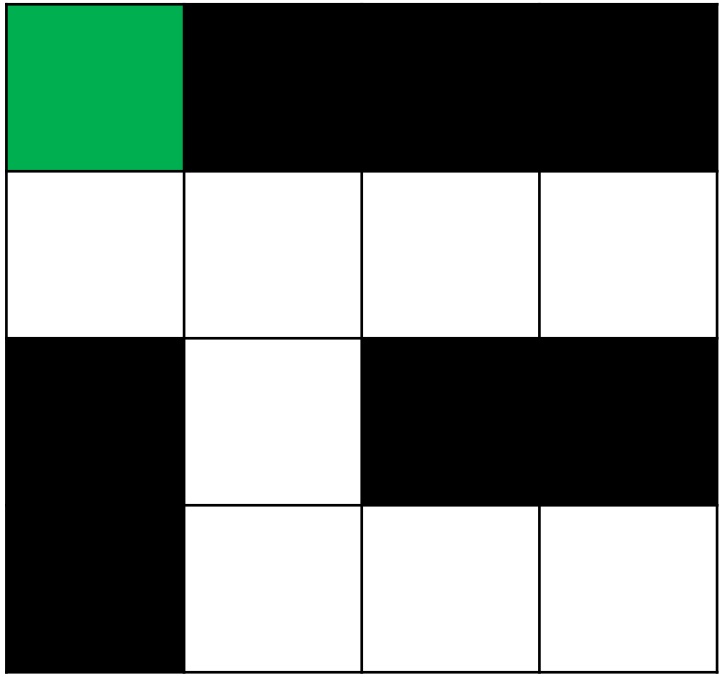
老鼠走迷宮

Q：假設走的先後順序依序是：右邊->左邊->下面->上面
這張3*3的地圖會怎麼走？



老鼠走迷宮

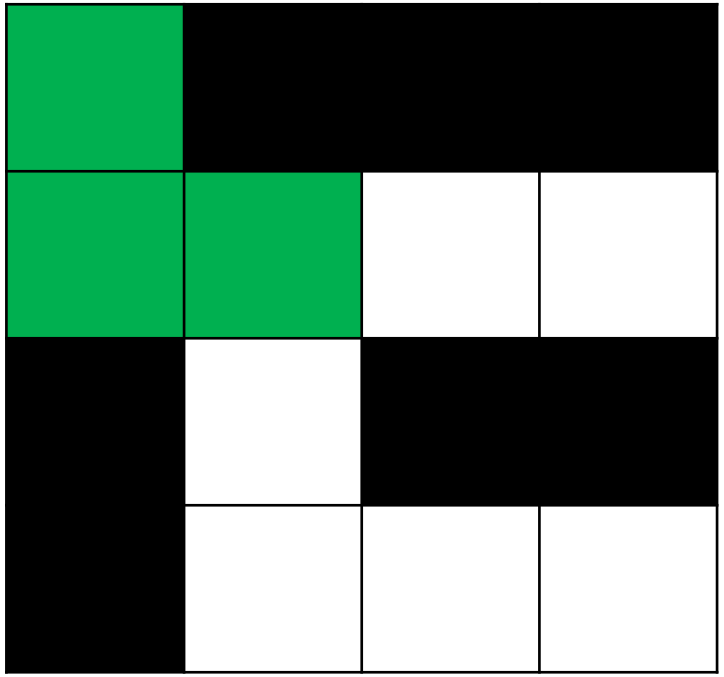
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(X)
2.向左(X)
3.向下(O)

老鼠走迷宮

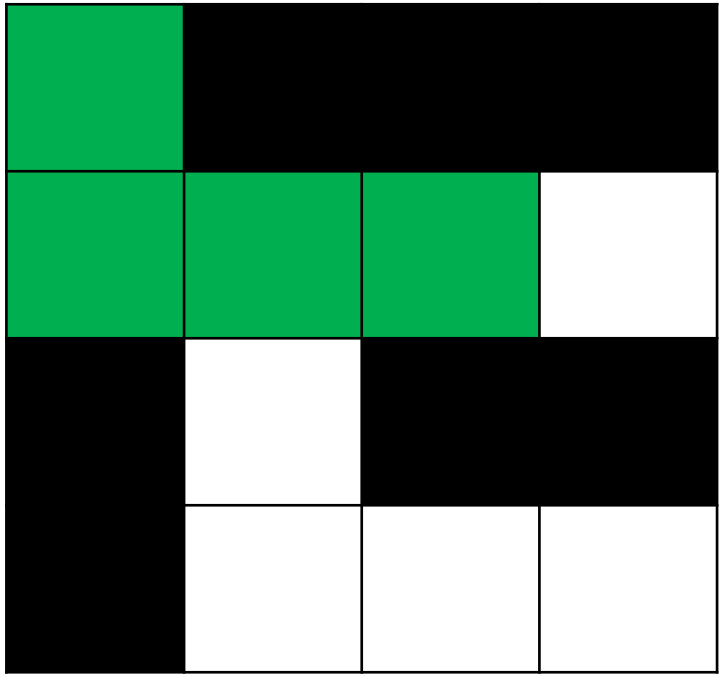
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(O)

老鼠走迷宮

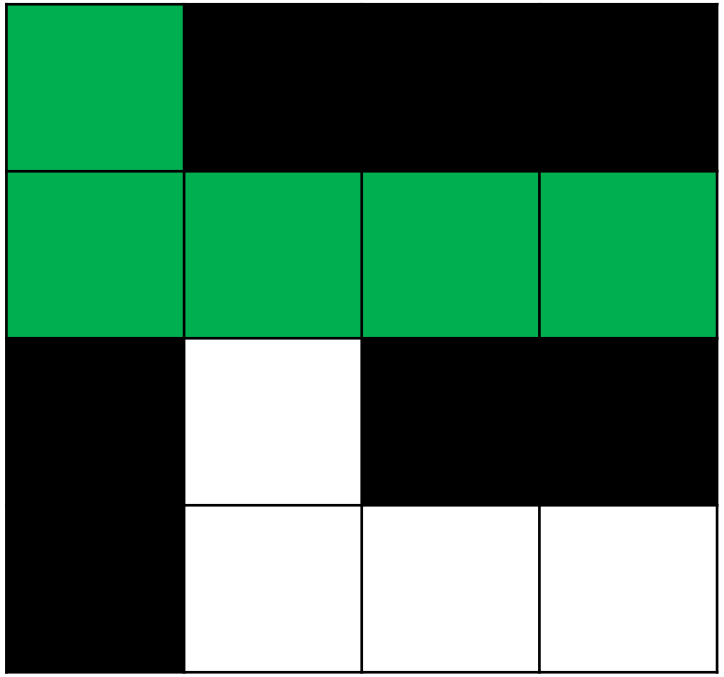
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(O)

老鼠走迷宮

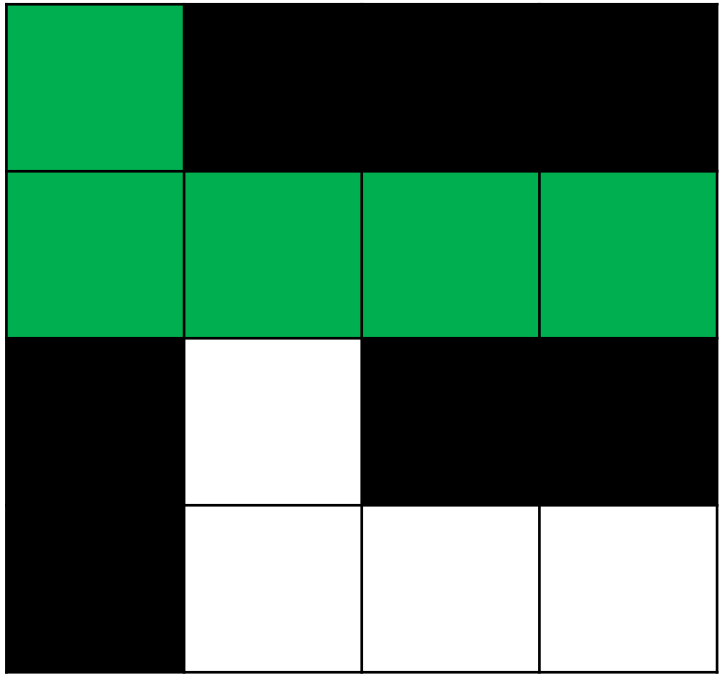
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(O)

老鼠走迷宮

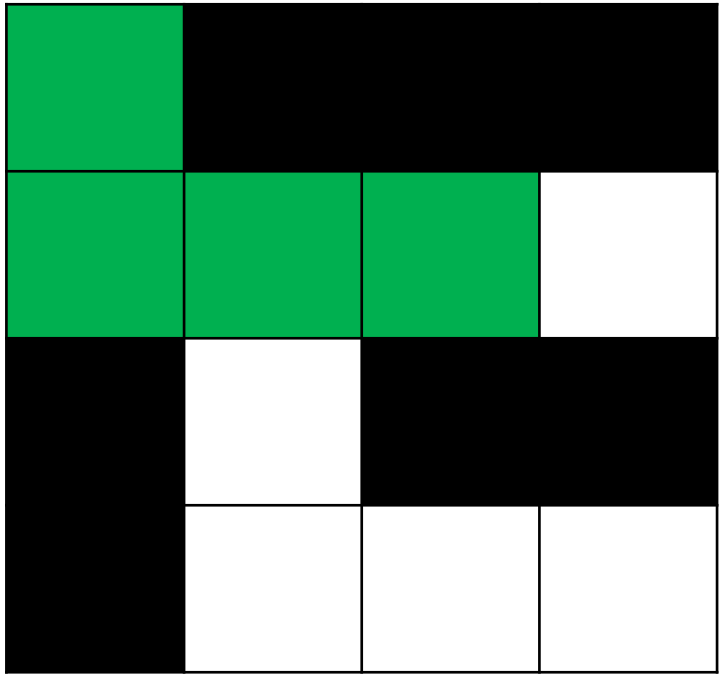
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(X)
2.向左(X)
3.向下(X)
4.向上(X)
都不能走了怎麼辦?
=>代表這一步是錯的，退回前一步

老鼠走迷宮

Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？

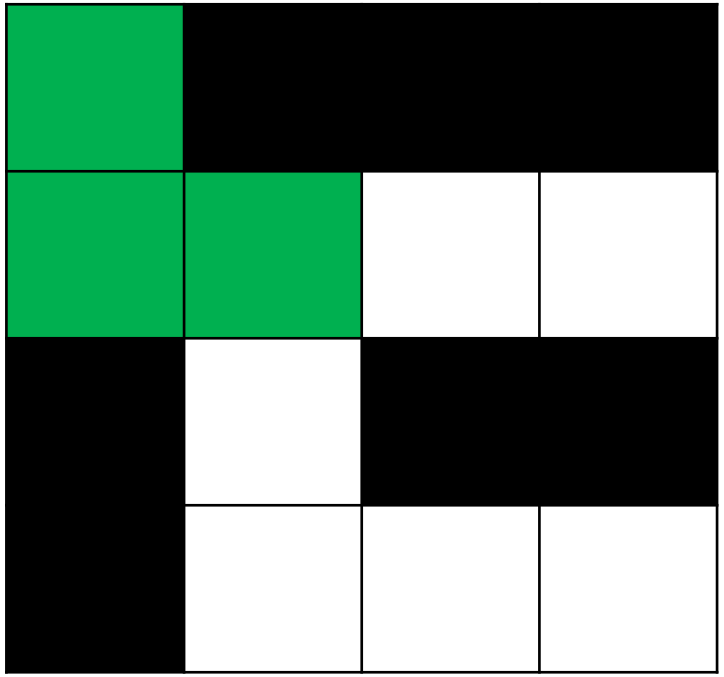


~~1.向右(O)~~
2.向左(X)
3.向下(X)
4.向上(X)

=>退回前一步

老鼠走迷宮

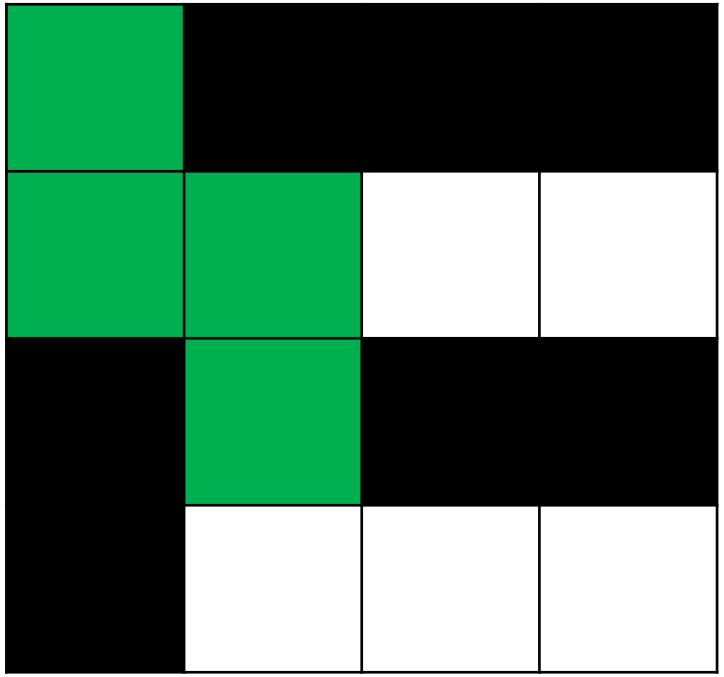
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



~~1.向右(O)~~
2.向左(X)
3.向下(O)

老鼠走迷宮

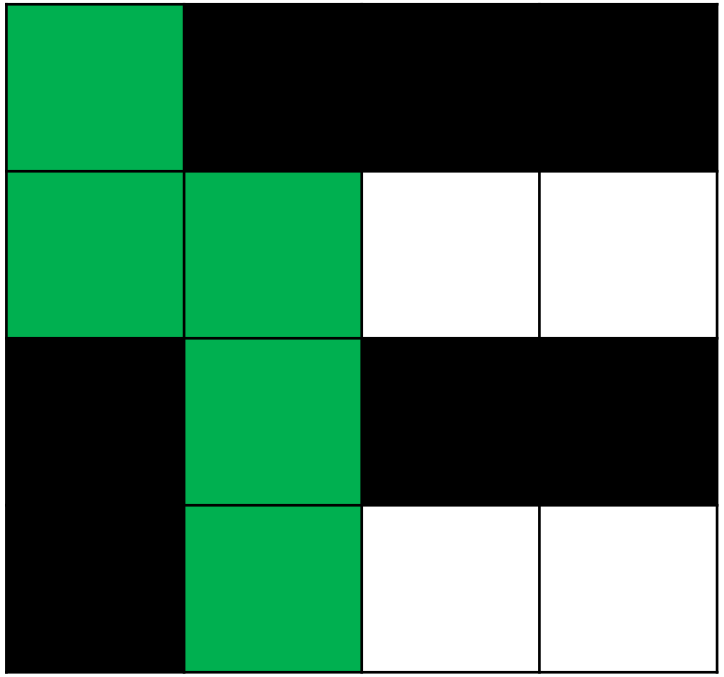
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(X)
2.向左(X)
3.向下(O)

老鼠走迷宮

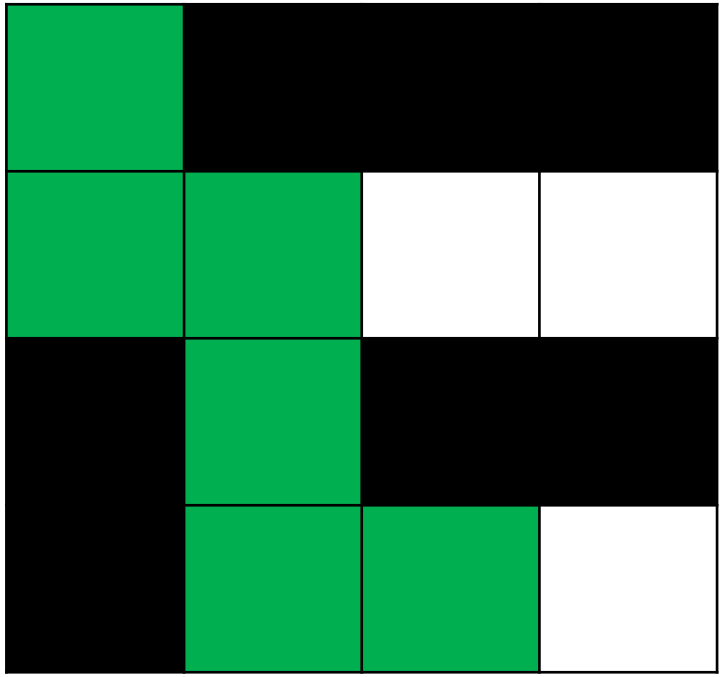
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(O)

老鼠走迷宮

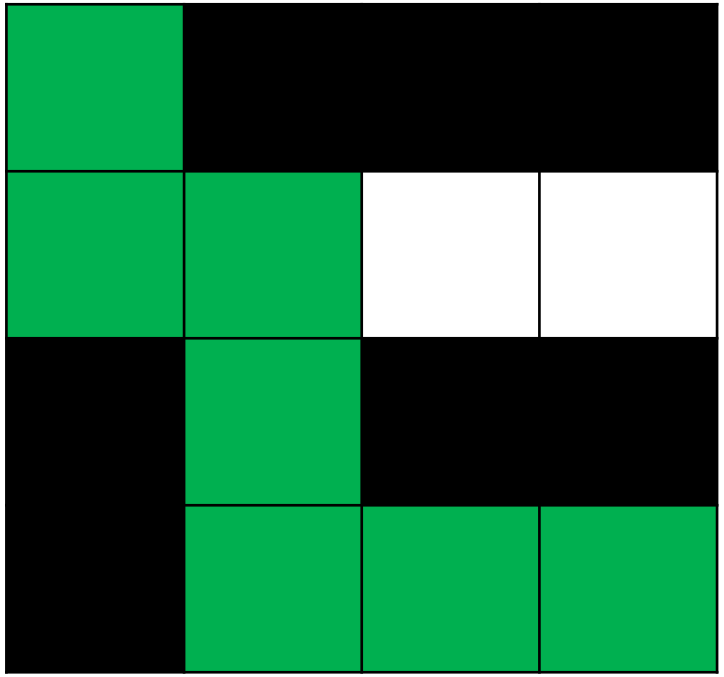
Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



1.向右(O)

老鼠走迷宮

Q2：假設走的先後順序依序是：右邊->左邊->下面->上面
這張4*4的地圖會怎麼走？



用stack的觀點來看老鼠走迷宮

Q：下面stack狀態中，假設(0,3)已經沒有路了怎麼辦?
A：pop掉top，在push(0,2)向左的code

(0,2)向左
(0,1)向右
(0,0)向右

走迷宮搜

設計 2 ~ 3 個你自己的地圖吧
（第一列的兩個數字是行跟列大小）
（數字用空格隔開）
設計完先放在筆記本裡

未命名 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
10 10
0 0 1 1 1 1 1 1 1 1
1 0 1 0 1 1 1 0 0 1
1 0 0 0 0 0 1 0 1 1
1 1 1 1 1 0 0 0 1 1
1 1 0 0 0 0 1 1 1 1
1 0 0 1 0 1 1 1 0 0
1 1 1 1 0 0 0 0 0 1
1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 1 0 0 1 1
1 1 1 1 1 1 1 0 0 0

4 4
0 0 1 1
1 0 0 1
1 1 0 0
1 1 1 0

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

走迷宮搜

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 100
int col,row;
int map[SIZE][SIZE] ={0};
bool is_end = false;
//2:走過的路線 1:牆壁 0:走道
int main()
{

    scanf("%d %d",&col,&row);
    for(int i = 0 ; i < col ;++i)
    {
        for(int j = 0 ; j < row ; ++j)
        {
            scanf("%d",&map[i][j]);
        }
    }
    int start_x = 0,start_y = 0;
    int end_x = col-1,end_y = row-1;
    visit(start_x,start_y,end_x,end_y);
    system("PAUSE");
}
```

走迷宮搜

```
void print_map()
{
    system("cls");
    for(int i = 0 ; i < col ;++i)
    {
        for(int j = 0 ; j < row ; ++j)
        {
            switch(map[i][j])
            {
                case 0 :printf("  ");break;
                case 1 :printf("■");break;
                case 2 :printf("* ");break;
            }
        }
        printf("\n");
    }
    _sleep(100);//延遲0.1秒
}
```

走迷宮搜

```
void visit(int x,int y ,int end_x,int end_y)
{
    print_map();//每次走前先印出地圖目前狀態
    if(x < 0 || x >= row || y < 0 || y >= col)return;//超過地圖大小就返回
    if(x == end_x && y == end_y)//走到終點
    {
        map[x][y] = 2;
        is_end = true;
        return;
    }
    if(is_end == true)return;//已經到終點了就返回
    if(map[x][y] == 0)
    {
        map[x][y] = 2;
        visit(x,y+1,end_x,end_y);
        visit(x,y-1,end_x,end_y);
        visit(x+1,y,end_x,end_y);
        visit(x-1,y,end_x,end_y);
    }
}
```

還是看不懂怎麼執行的？

善用中斷點跟Ｆ１０來看當下 x 跟 y 的狀態

input.c　new.cpp　10_5.cpp

(全域範圍)　visit(int x, int y, int end_x, int end_y)

```
24      }
25      _sleep(100);//延遲0.1秒
26  }
27  void visit(int x,int y ,int end_x,int end_y)
28  {
29      print_map();//每次走前先印出地圖目前狀態
30      if(x < 0 || x >= row || y < 0 || y >= col)return;//超過地圖大小就返回
31      if(x == end_x && y == end_y)//走到終點
32      {
33          map[x][y] = 2;
34          is_end = true;
35          return;
36      }
37      if(is_end == true)return;//已經到終點了就返回
38      if(map[x][y] == 0)
39      {
40          map[x][y] = 2;
41          visit(x,y+1,end_x,end_y);
42          visit(x,y-1,end_x,end_y);
43          visit(x+1,y,end_x,end_y);
```

100 %

區域變數

名稱	值	
x	2	
y	1	
end_x	3	
end_y	3	int

C:\Users\user\Desktop\10_5\Debug\10_5.exe

```
* *
 * *
  * *
   *
```

微軟注音 半 ：