Knight’s Tour

a.程式架構

b的部分就是先將起點輸入進去，接著程式就會開始跑，會判斷下一步哪個能跑哪個不能跑，當所有路線都不能跑時，迴圈會結束，在來把棋盤上的數字列印出來。

c的部分比b還要複雜，先判斷下一步有哪幾種走法能走，用陣列儲存座標，再看下一步的下一步有多少走法（也是用陣列儲存），找最少的，有最少的下一步的下一步就是我們要的答案。接著程式會把棋盤列印出來。

附帶一提，棋盤上沒走到的位置用０表示。

（圖片取動朋的Ｋ島）

ｂ.討論

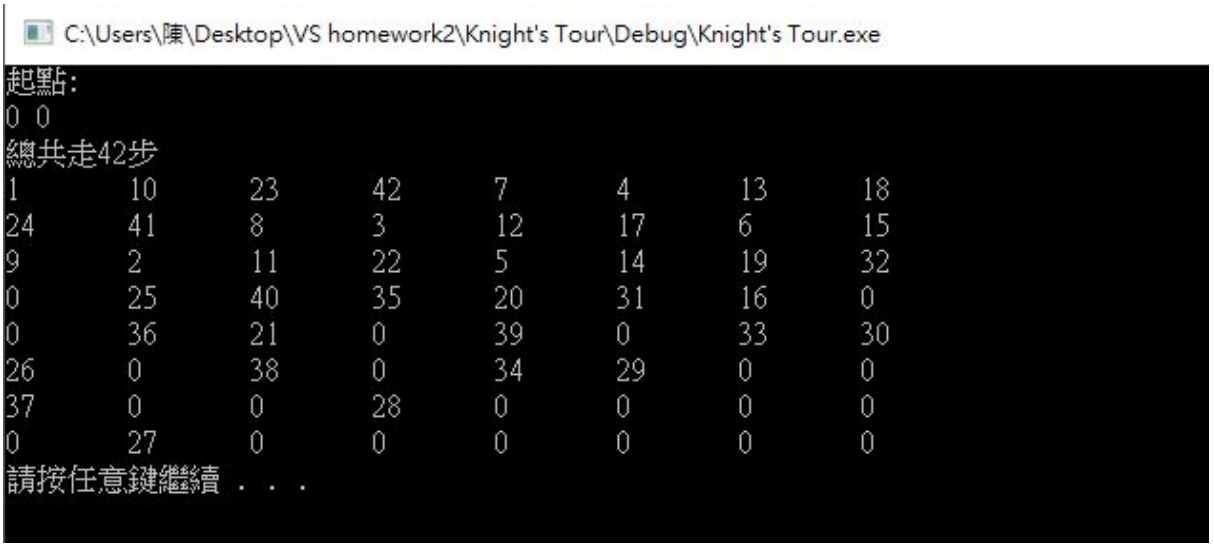
不管是哪一題都困擾了我很久，但努力是值得的。

但討論的部分我不想說我是怎麼想出程式來的，因為那就跟一個靈感像風一樣吹進腦海裡是類似的。

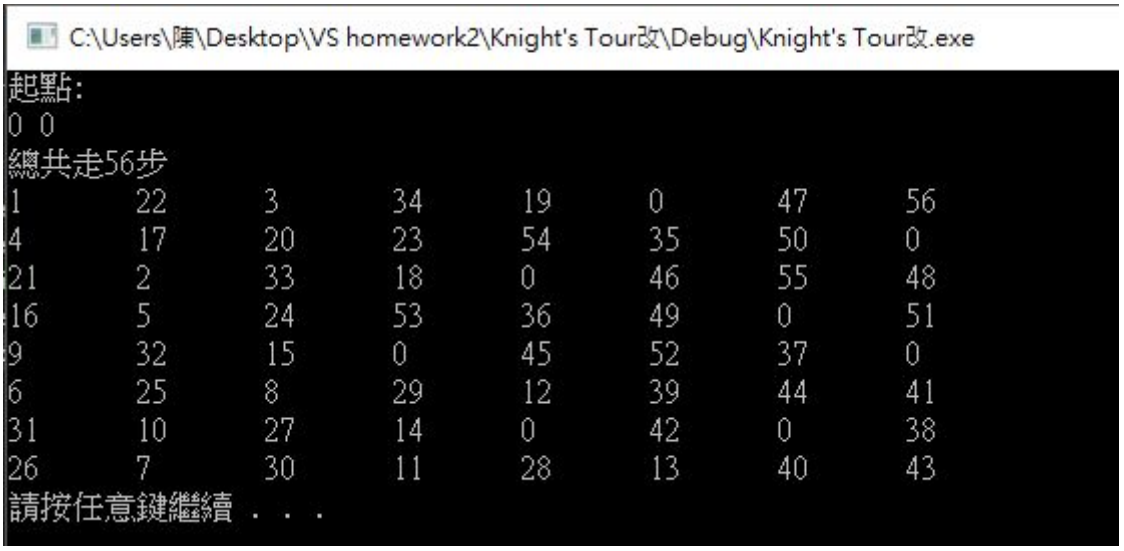
我在打報告時（對，就是現在），還上網找關於knight's tour比較有趣的事，除了土耳其行棋傀儡外沒其他的，沒有河內塔那樣的~~，在西洋棋尋求世界末日是否搞錯子什麼~~。

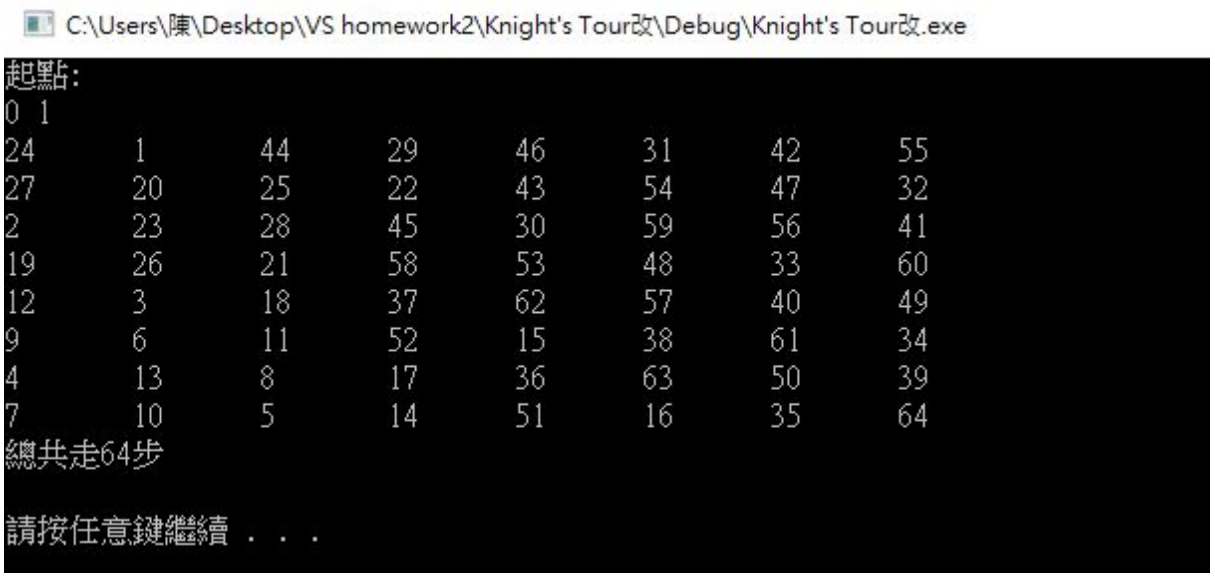
c.執行畫面

b


C:\Users\陳\Desktop\VS homework2\Knight's Tour\Debug\Knight's Tour.exe
起點:
0 0
總共走42步
1 10 23 42 7 4 13 18
24 41 8 3 12 17 6 15
9 2 11 22 5 14 19 32
0 25 40 35 20 31 16 0
0 36 21 0 39 0 33 30
26 0 38 0 34 29 0 0
37 0 0 28 0 0 0 0
0 27 0 0 0 0 0 0
請按任意鍵繼續 . . .

c


C:\Users\陳\Desktop\VS homework2\Knight's Tour改\Debug\Knight's Tour改.exe
起點:
0 0
總共走56步
1 22 3 34 19 0 47 56
4 17 20 23 54 35 50 0
21 2 33 18 0 46 55 48
16 5 24 53 36 49 0 51
9 32 15 0 45 52 37 0
6 25 8 29 12 39 44 41
31 10 27 14 0 42 0 38
26 7 30 11 28 13 40 43
請按任意鍵繼續 . . .



d.程式碼

b.

```
# include<stdio.h>
# include<stdlib.h>

# define SIZE 8

int hori[8] = { 2, 1, -1, -2, -2, -1, 1, 2 },
    vert[8] = { -1, -2, -2, -1, 1, 2, 2, 1 };

void knightour(int a[][8], size_t l, int x, int y);

int main(){

    int currentrow, currentclumn;

    int position[SIZE][SIZE] = { 0 };

    puts("起點:");

    scanf("%d", &currentrow);
```

```
	scanf("%d", &currentclumn);

	knightour(position, SIZE, currentclumn, currentrow);

	for (size_t y = 0; y < SIZE; y++){

		for (size_t x = 0; x < SIZE; x++){

			printf("%d   ", position[y][x]);
		}

		puts("");
	}

	system("pause");
}

void knightour(int a[][8], size_t l, int x, int y){

	a[y][x] = 1;

	for (int i = 2; i < 65; i++){

		if (a[y + vert[0]][x + hori[0]] == 0 && y + vert[0] >= 0 && y +
vert[0] < 8 && x + hori[0] >= 0 && x + hori[0] < 8){

			x += hori[0];
			y += vert[0];

			a[y][x] = i;
		}

		else if (a[y + vert[1]][x + hori[1]] == 0 && y + vert[1] >= 0 && y
+ vert[1] < 8 && x + hori[1] >= 0 && x + hori[1] < 8){
```

```
                x += hori[1];
                y += vert[1];

                a[y][x] = i;
            }

            else if (a[y + vert[2]][x + hori[2]] == 0 && y + vert[2] >= 0 && y
+ vert[2] < 8 && x + hori[2] >= 0 && x + hori[2] < 8){

                x += hori[2];
                y += vert[2];

                a[y][x] = i;
            }

            else if (a[y + vert[3]][x + hori[3]] == 0 && y + vert[3] >= 0 && y
+ vert[3] < 8 && x + hori[3] >= 0 && x + hori[3] < 8){

                x += hori[3];
                y += vert[3];

                a[y][x] = i;
            }

            else if (a[y + vert[4]][x + hori[4]] == 0 && y + vert[4] >= 0 && y
+ vert[4] < 8 && x + hori[4] >= 0 && x + hori[4] < 8){

                x += hori[4];
                y += vert[4];

                a[y][x] = i;
            }

            else if (a[y + vert[5]][x + hori[5]] == 0 && y + vert[5] >= 0 && y
+ vert[5] < 8 && x + hori[5] >= 0 && x + hori[5] < 8){
```

```
                x += hori[5];
                y += vert[5];

                a[y][x] = i;
            }

            else if (a[y + vert[6]][x + hori[6]] == 0 && y + vert[6] >= 0 && y
+ vert[6] < 8 && x + hori[6] >= 0 && x + hori[6] < 8){

                x += hori[6];
                y += vert[6];

                a[y][x] = i;
            }

            else if (a[y + vert[7]][x + hori[7]] == 0 && y + vert[7] >= 0 && y
+ vert[7] < 8 && x + hori[7] >= 0 && x + hori[7] < 8){

                x += hori[7];
                y += vert[7];

                a[y][x] = i;
            }

            else{

                printf("總共走%d步\n", --i);

                break;
            }
        }
}

        c
```

```
# include<stdio.h>
# include<stdlib.h>

# define SIZE 8

int hori[8] = { 2, 1, -1, -2, -2, -1, 1, 2 },
    vert[8] = { -1, -2, -2, -1, 1, 2, 2, 1 };

void knightour(int a[][8], size_t l, int x, int y);

int main(){

        int currentrow, currentclumn;

        int position[SIZE][SIZE] = { 0 };

        puts("起點:");

        scanf("%d", &currentrow);
        scanf("%d", &currentclumn);

        knightour(position, SIZE, currentclumn, currentrow);

        for (size_t y = 0; y < SIZE; y++){

                for (size_t x = 0; x < SIZE; x++){

                        printf("%d   ", position[y][x]);

                        if (position[y][x] == 64){

                                puts("");
                                puts("總共走64步");
                        }
                }
```

```
                puts("");

            }

            system("pause");
        }

        void knightour(int a[][8], size_t l, int x, int y){

            a[y][x] = 1;

            for (int k = 2; k < 65; k++){

                int flag[8] = { 0 };

                int next_x[8] = { 0 }, next_y[8] = { 0 };

                int next_next_step[8] = { 8 };

                for (int i = 0; i < 8; i++){

                    if (a[y + vert[i]][x + hori[i]] == 0 && y + vert[i] >= 0
&& y + vert[i] < 8 && x + hori[i] >= 0 && x + hori[i] < 8){

                        flag[i] = 1;
                    }
                }

                if (flag[0] != 1 && flag[1] != 1 && flag[2] != 1 && flag[3]
!= 1 && flag[4] != 1 && flag[5] != 1 && flag[6] != 1 && flag[7] != 1){

                    printf("總共走%d步\n", --k);

                    break;
```

```
                }
                else{

                        for (int i = 0; i < 8; i++){

                                if (flag[i] == 1){

                                        next_x[i] = x + hori[i];
                                        next_y[i] = y + vert[i];
                                }
                        }

                        for (int i = 0; i < 8; i++){

                                if (flag[i] == 1){

                                        for (int j = 0; j < 8; j++){

                                                if (next_x[i] + hori[j] < 0 ||
next_x[i] + hori[j]>7 || next_y[i] + vert[j] < 0 || next_y[i] + vert[j]>7 ||
a[next_y[i] + vert[j]][next_x[i] + hori[j]] != 0){

                                                        next_next_step[i]--;
                                                }
                                        }

                                }

                        }

                        int mini = next_next_step[0];

                        for (int i = 1; i < 8; i++){

                                if (flag[i] == 1){
```

```
                        if (mini>next_next_step[i]){

                            mini = next_next_step[i];
                        }
                    }
                }

                int next_type;

                for (int i = 0; i < 8; i++){

                    if (flag[i] == 1){

                        if (next_next_step[i] == mini){

                            next_type = i;

                            break;
                        }
                    }
                }

                x += hori[next_type];
                y += vert[next_type];

                a[y][x] = k;
            }
        }
    }
```