

程式設計(一)期末專題 象棋程式

學生：1062023 陳宣妤、1062035 曾羽羚
指導教授：方文聘

中華民國 109 年 01 月 06 日

目 錄

壹、問題說明.....	2
貳、演算法	2
一、棋子移動的順序	2
二、將的防守	2
三、棋子的移動方式	2
參、程式	4
一、流程圖	4
.....	5
.....	6
肆、討論	7
一、功能	7
二、缺陷	7
三、問題與解決方法	7
陸、附錄	8
一、程式碼	8

壹、問題說明

此次專題是象棋對戰，只需吃掉對方的王就獲勝，如果雙方的王在規定步數內皆沒有死亡，則比較雙方吃掉的棋子數量。雖然勝利看似簡單，但每顆棋移動方式都不同，因此我們需要先分析我們每顆棋的行走路徑，考慮移動的方式與規則，思考進攻防守策略。我們主要的想法是先防禦王的四周，再利用車能夠在沒有其他棋子阻擋的情況下每一步可以沿著直線橫走或直走任意步，步數不限，不可斜走或拐彎的特性進行主要進攻。積極進攻獲勝機率雖然高，倘若每個玩家的想法皆相同或遇到會防守的玩家，卻很容易因為只專注於進攻讓自己的王被吃掉而輸了棋局。因此優先判斷王的防守，再積極進攻是我們認為較好的方法。而我們可能會遇到不少問題，像是如何遵守每顆棋移動方式的限制？如何判斷可以吃棋？如何判斷棋盤的邊界？如何計算棋子的路徑？如何找尋對方王的位置？這些都是我們將想法與策略轉換成程式上可能會遇到的問題，也是我們經過多次討論與嘗試解決讓程式變得更完善的原因。

貳、演算法

一、棋子移動的順序

先給每顆棋子分數，要先動的棋子分數越高，以此類推，分數越低的就越慢動到，分數為 0~16。將所有棋子照分數順序由大到小排列，被吃掉的棋子不列入算分，所以 $P[0]$ 的分數永遠最高，再用 $P[0]$ 的 id 來判斷它是什麼棋子，接著採用對應的移動方式。

二、將的防守

在判斷分數之前，先判斷將的前後左右是否有對方的棋子，若有則吃掉，沒有則往下判斷其他棋子的移動。

三、棋子的移動方式

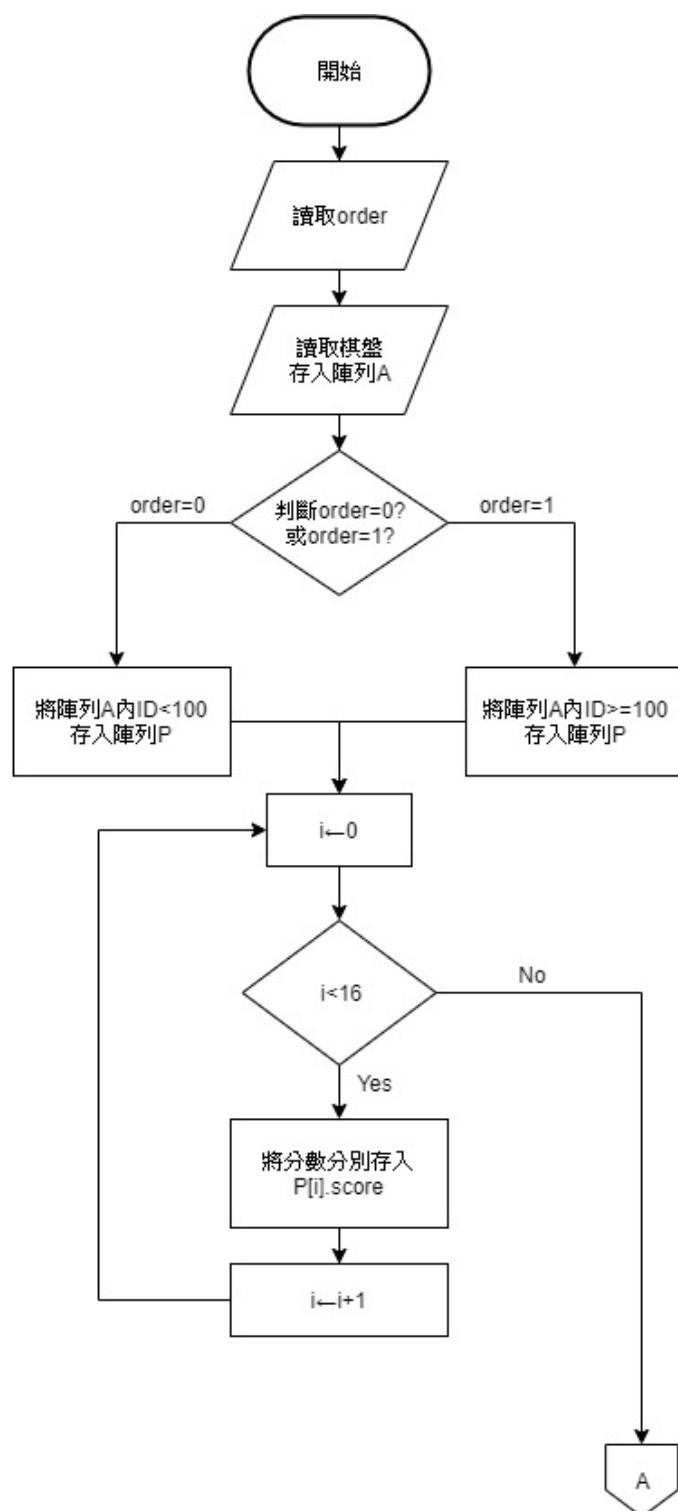
1. 車：最先移動(分數 16、15)。先分別計算上下左右到碰到其他棋子或邊界的長度為多少後，判斷碰到的是否為對方的王，是則移動車吃王，否則判斷碰到的是否為對方的棋子，若判定為對方的棋子，則移動車吃棋，否則亂數下棋在計算最長的路徑內。
2. 砲：若車都死光，接著移動砲(分數 14、13)。分別計算上下左右路徑上是否有超過兩顆棋子，若有且第二顆為對方的棋子則移動砲吃棋，若路徑上低於

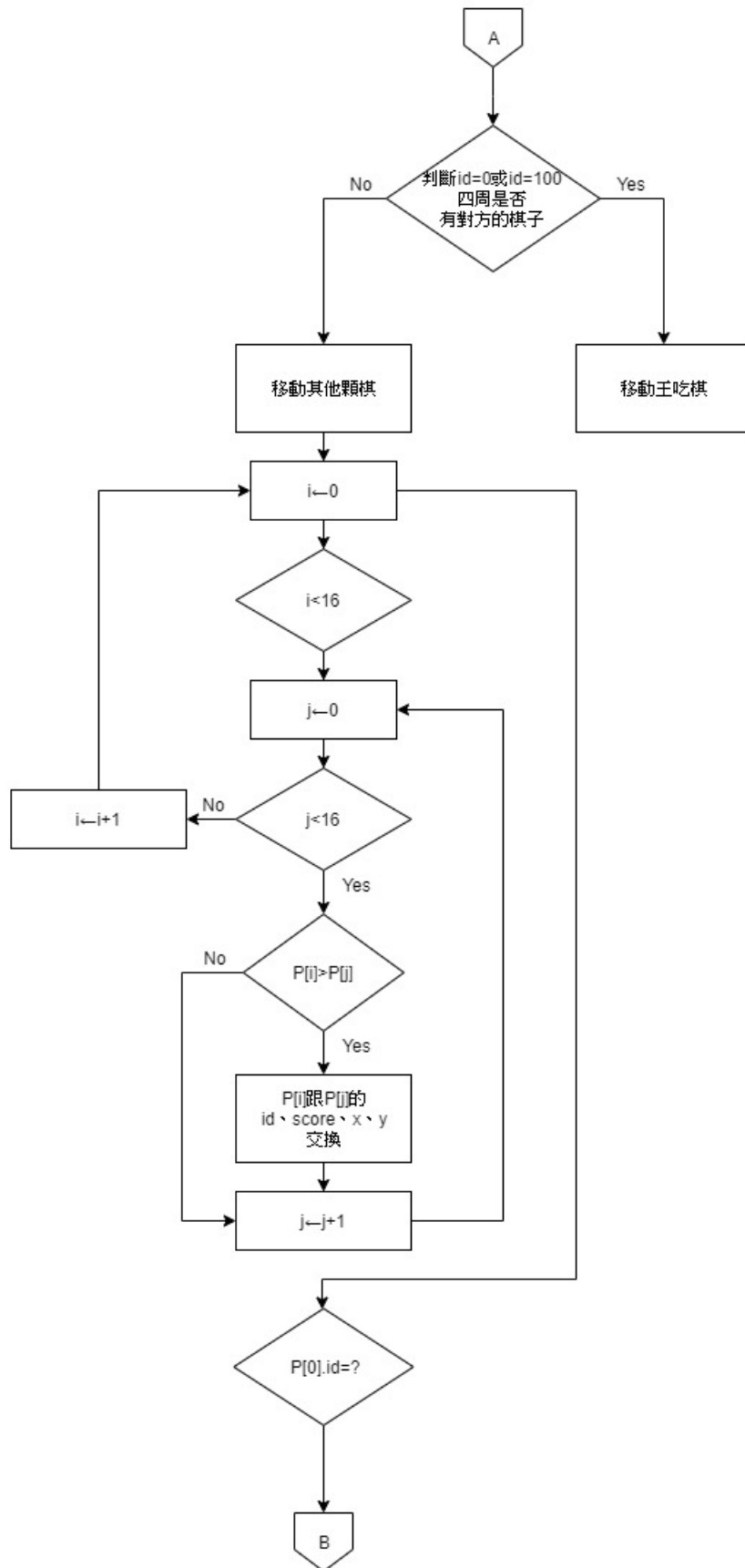
兩顆棋子或第二顆棋為自己的，計算上下左右的路徑長度，亂數下棋在計算的最長的路徑內。

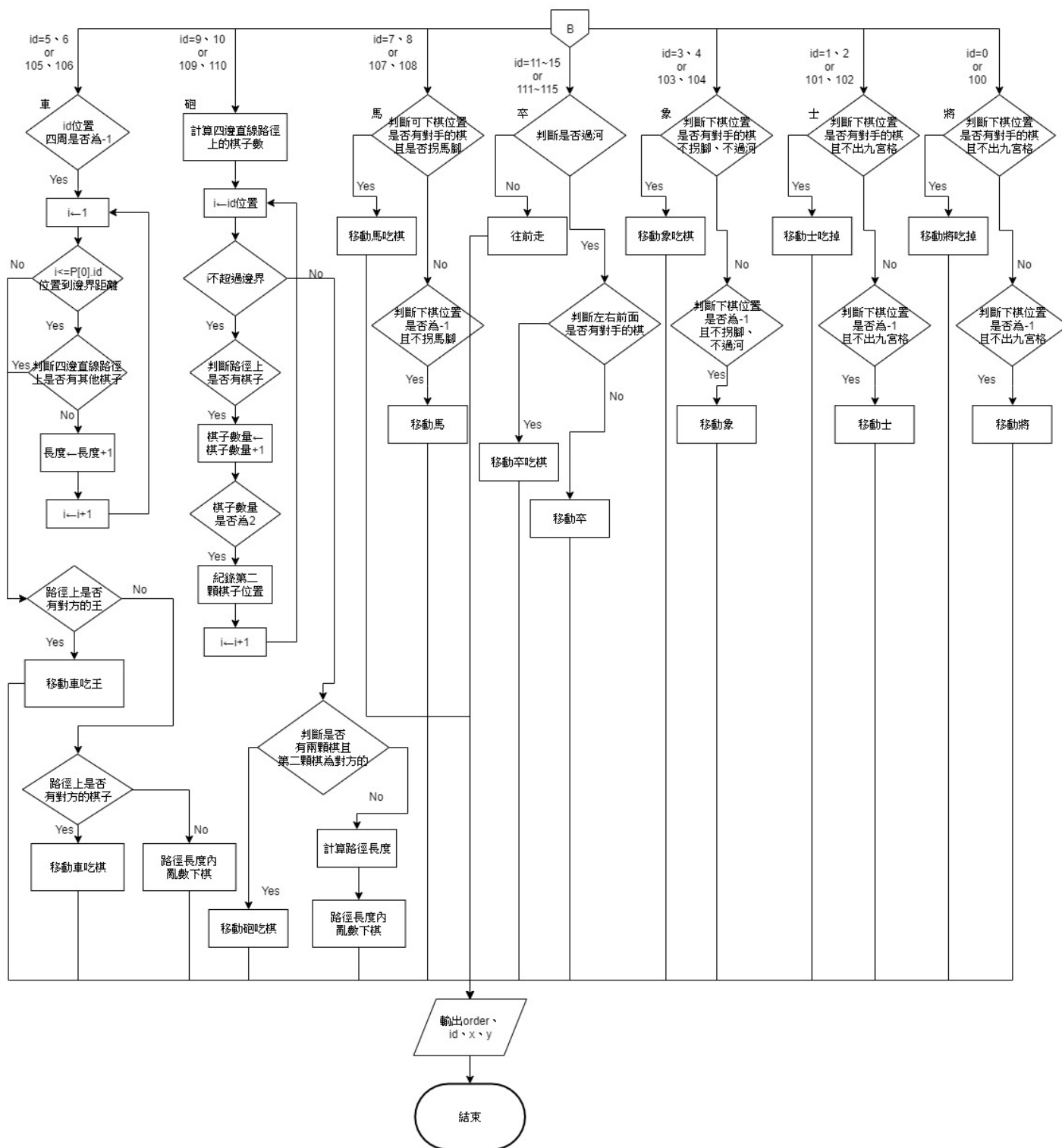
3. 馬：車跟砲皆死光，移動馬(分數 12、11)。判斷可能走的八種位置是否會拐馬腳、不超過邊界且有對方的棋子，如果有對方的棋子移動馬吃棋，沒有則根據判斷的位置下棋。
4. 卒：車、砲、馬皆死光，移動卒(分數 10、9、8、7、6)。過河前只能往前走，過河後判斷左右是否有對方的棋，有則移動卒吃棋，沒有則繼續往下走。
5. 象：車、砲、馬、卒皆死光，移動象(分數 4、3)。判斷可能走的四種位置是否會拐腳、不過河、不超過邊界且有對方的棋子，如果有對方的棋子一動象吃棋，沒有則根據判斷的位置下棋。
6. 士：若只剩士跟將，先移動士(分數 2、1)。判斷九宮格內可走的位置、不超過邊界且有對方的棋子，如果有移動士吃棋，沒有則根據判斷的位置下棋。
7. 將：當最後只剩下將時，判斷前後左右是否有對方的棋子，有則移動將吃棋，沒有則根據判斷的位置下棋。

參、程式

一、流程圖







肆、討論

一、功能

1. 根據我們想要的順序先後移動棋子。
2. 所有棋子在移動前都會先判斷是否可以吃棋。
3. 車跟砲會根據計算的長度移動。
4. 車會最先判斷是否能吃將，是的話優先吃將。

二、缺陷

1. 沒有特別戰術，輸贏碰運氣。
2. 只有將會防守。
3. 一次只會動同一顆棋子，被吃掉才會動另外一顆棋子。
4. 車跟砲容易在直線上亂衝。

三、問題與解決方法

1. 一開始想用棋盤存入分數，但是我們不知道如何運用通用的方式計算分數，而且不知道如何在棋盤中存入分數，因此我們改用給每顆棋子分數的方式。
2. 原本希望車在第二步的時候走到指定的位置上，但是這樣無法靈活運用車的特性，可能會變成一步一步走，因此改用在直線距離上亂數的方式移動。
3. 原本砲是路徑上有一顆棋就繼續往後找有沒有第二顆，再判斷第二顆棋是否為對方的，若這樣沒有找到第二顆棋，迴圈無法中斷。所以我們改用先找路徑上的第二顆棋，將位置記錄起來，再另外判斷他是否是別人的棋。
4. 紅棋的吃棋判斷一開始是寫 <100 ，但是 -1 也包含在 <100 的範圍內，這樣無論棋盤上是空的還是黑棋，都會移動到該位子，因此我們加上 ≥ 0 的判斷解決這個問題。
5. 一開始黑棋是判斷上下左右是否 ≥ 100 ，若棋子在 $(0,0)$ ，可能會往 -1 判斷到棋盤外，且會判斷成 ≥ 100 ，而超出棋盤，因此我們在每條判斷式中都加上邊界限制。

6. 車的移動方向判定方式是往直線距離最長的方向走，一開始是讓車往該方向長度>其他方向長度時先移動，倘若遇到長度相同時，會無法判斷，而停在原地，因此改用該方向長度>=其他方向時車便可以順利移動。
7. 原本車是在計算的長度範圍內亂數移動，可是這容易造成車在卒跟原本的位置來回移動而出不去，因此我們將亂數範圍改成長度-1，讓車不會直接衝到底。

伍、參考資料

1. 象棋規則：

<http://www.cccs.org.tw/front/bin/ptdetail.phtml?Part=601&Category=100177>

陸、附錄

一、程式碼

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <stdlib.h>
#include <stdlib.h>
#include <time.h>
#define W 9
#define H 10
using namespace std;
struct piece
{
    int id; //棋子代表數字
    int x; //位置
    int y;
    int score = 0;
};
int main(int n, char** argv)
{
    srand((unsigned)time(NULL));

    int order;
    if (n >= 2) order = atoi(argv[1]);
    cout << order << endl;
    int A[W][H] = { 0 };

    /*-----Read table-----*/
    ifstream tablefile("table.txt");

    int i, j;
    for (i = 0; i < H; i++)
```

```

{
    for (j = 0; j < W; j++)
    {
        tablefile >> A[j][i];    //從檔案輸入，讀檔案內容
    }
}

tablefile.close();

for (i = 0; i < H; i++)
{
    for (j = 0; j < W; j++)
    {
        cout << setw(4) << A[j][i];
    }
    cout << endl;
}

/*-----目前的位置-----*/

struct piece P[16] = { 0 };    //16 顆棋
int m = 0;                    //存現有棋子

for (i = 0; i < H; i++)
{
    for (j = 0; j < W; j++)
    {
        if (A[j][i] >= 0)
        {
            if (order == 0)
            {
                if (A[j][i] < 100)
                {
                    P[m].id = A[j][i];
                    P[m].x = j;
                    P[m].y = i;
                    m++;
                }
            }
            else if (order == 1)
            {
                if (A[j][i] >= 100)
                {
                    P[m].id = A[j][i];
                    P[m].x = j;
                    P[m].y = i;
                    m++;
                }
            }
        }
    }
}

```

```

    }
}
for (i = 0; i < m; i++)

    cout << P[i].id << ":" << "(" << P[i].x << "," << P[i].y << ")" << endl;
/*-----下棋位置-----*/
int id = 0, x = 0, y = 0;
/*-----給每顆棋子分數-----*/
for (int i = 0; i < 16; i++) {
    if (P[i].id == 6 || P[i].id == 106) //車 1
    {
        P[i].score = 16;
    }
    else if (P[i].id == 5 || P[i].id == 105) //車 2
    {
        P[i].score = 15;
    }
    else if (P[i].id == 3 || P[i].id == 103) //象 1
    {
        P[i].score = 3;
    }
    else if (P[i].id == 4 || P[i].id == 104) //象 2
    {
        P[i].score = 4;
    }
    else if (P[i].id == 0 || P[i].id == 100) //將
    {
        P[i].score = 0;
    }
    else if (P[i].id == 1 || P[i].id == 101) //士 1
    {
        P[i].score = 1;
    }
    else if (P[i].id == 2 || P[i].id == 102) //士 2
    {
        P[i].score = 2;
    }
    else if (P[i].id == 7 || P[i].id == 107) //馬 1
    {
        P[i].score = 12;
    }
    else if (P[i].id == 8 || P[i].id == 108) //馬 2
    {
        P[i].score = 11;
    }
    else if (P[i].id == 9 || P[i].id == 109) //砲 1
    {

```

```

        P[i].score = 14;
    }
    else if (P[i].id == 10 || P[i].id == 110) //砲 2
    {
        P[i].score = 13;
    }
    else if (P[i].id == 11 || P[i].id == 111) //卒 2
    {
        P[i].score = 7;
    }
    else if (P[i].id == 12 || P[i].id == 112) //卒 3
    {
        P[i].score = 9;
    }
    else if (P[i].id == 13 || P[i].id == 113) //卒 1
    {
        P[i].score = 10;
    }
    else if (P[i].id == 14 || P[i].id == 114) //卒 4
    {
        P[i].score = 8;
    }
    else if (P[i].id == 15 || P[i].id == 115) //卒 5
    {
        P[i].score = 6;
    }
}
/*-----黑棋-----*/
if (order == 0)
{
    //將的防守
    int king = 0;
    for (int i = 0; i < 16; i++)
    {
        if (P[i].id == 0)
        {
            king = i;
            break;
        }
    }
    if (A[P[king].x + 1][P[king].y] >= 100 && P[king].x + 1 <= 5) //將的正右
方有東西吃
    {
        x = P[king].x + 1;
        y = P[king].y;
        P[0].id = 0;
    }
    else if (A[P[king].x - 1][P[king].y] >= 100 && P[king].x - 1 >= 3) //將的正左

```

```

方有東西吃
{
    x = P[king].x - 1;
    y = P[king].y;
    P[0].id = 0;
}
else if (A[P[king].x][P[king].y + 1] >= 100 && P[king].y + 1 <= 2) //將的正
下方有東西吃
{
    x = P[king].x;
    y = P[king].y + 1;
    P[0].id = 0;
}
else if (A[P[king].x][P[king].y - 1] >= 100 && P[king].y - 1 >= 0) //將的正上
方有東西吃
{
    x = P[king].x;
    y = P[king].y - 1;
    P[0].id = 0;
}
//若將周圍沒有對手棋
else
{
    //分數排序
    for (int i = 0; i < 16; i++) {
        for (int j = 0; j < 16; j++) {
            if (P[i].score > P[j].score) {
                int tmp = P[i].score, tmp2 = P[i].id, tmp3 =
P[i].x, tmp4 = P[i].y;
                P[i].score = P[j].score;
                P[i].id = P[j].id;
                P[i].x = P[j].x;
                P[i].y = P[j].y;
                P[j].score = tmp;
                P[j].id = tmp2;
                P[j].x = tmp3;
                P[j].y = tmp4;
            }
        }
    }
    int len1 = 0; //左邊
    int len2 = 0; //右邊
    int len3 = 0; //後面
    int len4 = 0; //前面
    /*-----車-----*/
    if (P[0].id == 6 || P[0].id == 5) {
        if (A[P[0].x - 1][P[0].y] == -1) { //若左邊可以走，計算長度
            for (int i = 1; i <= P[0].x; i++) {

```

```

        if (A[P[0].x - i][P[0].y] == -1) {
            len1++;
        }
        else
            break;
    }
}

if (A[P[0].x + 1][P[0].y] == -1) { //若右邊可以走，計算長度
    for (int i = 1; i <= 8 - P[0].x; i++) {
        if (A[P[0].x + i][P[0].y] == -1) {
            len2++;
        }
        else
            break;
    }
}

if (A[P[0].x][P[0].y - 1] == -1) { //若上面可以走，計算長度
    for (int i = 1; i <= P[0].y; i++) {
        if (A[P[0].x][P[0].y - i] == -1) {
            len3++;
        }
        else
            break;
    }
}

if (A[P[0].x][P[0].y + 1] == -1) { //若下面可以走，計算長度
    for (int i = 1; i <= 9 - P[0].y; i++) {
        if (A[P[0].x][P[0].y + i] == -1) {
            len4++;
        }
        else
            break;
    }
}

//判斷車能否吃將
if (A[P[0].x][P[0].y + len4 + 1] == 100 && P[0].y + len4 + 1
<= 9) { //往下吃將
    x = P[0].x;
    y = P[0].y + len4 + 1;
}

else if (A[P[0].x - len1 - 1][P[0].y] == 100 && P[0].x - len1 - 1
>= 0) { //往左吃將
    x = P[0].x - len1 - 1;
    y = P[0].y;
}

else if (A[P[0].x + len2 + 1][P[0].y] == 100 && P[0].x + len2 +
1 <= 8) { //往右吃將
    x = P[0].x + len2 + 1;

```

```

        y = P[0].y;
    }
    else if (A[P[0].x][P[0].y - len3 - 1] == 100 && P[0].y - len3 - 1
    >= 0) { //往上吃將
        x = P[0].x;
        y = P[0].y - len3 - 1;
    }
    //若不是將，但為對手棋
    else if (A[P[0].x][P[0].y + len4 + 1] >= 100 && P[0].y + len4 +
    1 <= 9) { //往下吃
        x = P[0].x;
        y = P[0].y + len4 + 1;
    }
    else if (A[P[0].x - len1 - 1][P[0].y] >= 100 && P[0].x - len1 - 1
    >= 0) { //往左吃
        x = P[0].x - len1 - 1;
        y = P[0].y;
    }
    else if (A[P[0].x + len2 + 1][P[0].y] >= 100 && P[0].x + len2 +
    1 <= 8) { //往右吃
        x = P[0].x + len2 + 1;
        y = P[0].y;
    }
    else if (A[P[0].x][P[0].y - len3 - 1] >= 100 && P[0].y - len3 - 1
    >= 0) { //往上吃
        x = P[0].x;
        y = P[0].y - len3 - 1;
    }
    //都沒得吃，在路徑最長的方向亂數下棋
    else if (len1 >= len2 && len1 >= len3 && len1 >= len4) { //往
    左
        x = P[0].x - (rand() % (len1 - 1) + 1);
        y = P[0].y;
    }
    else if (len2 >= len1 && len2 >= len3 && len2 >= len4) //往右
    {
        x = P[0].x + (rand() % (len2 - 1) + 1);
        y = P[0].y;
    }
    else if (len3 >= len2 && len3 >= len1 && len3 >= len4) { //往
    上
        y = P[0].y - (rand() % (len3 - 1) + 1);
        x = P[0].x;
    }
    else if (len4 >= len2 && len4 >= len3 && len4 >= len1) { //往
    下
        y = P[0].y + (rand() % (len4 - 1) + 1);
        x = P[0].x;
    }

```

```

    }
}
/*-----砲-----*/
else if (P[0].id == 9 || P[0].id == 10) {
    if (A[P[0].x - 1][P[0].y] == -1) { //左邊可以走，計算長度
        for (int i = 1; i <= P[0].x; i++) {
            if (A[P[0].x - i][P[0].y] == -1) {
                len1++;
            }
            else
                break;
        }
    }
    if (A[P[0].x + 1][P[0].y] == -1) { //右邊可以走，計算長度
        for (int i = 1; i <= 8 - P[0].x; i++) {
            if (A[P[0].x + i][P[0].y] == -1) {
                len2++;
            }
            else
                break;
        }
    }
    if (A[P[0].x][P[0].y - 1] == -1) { //上面可以走，計算長度
        for (int i = 1; i <= P[0].y; i++) {
            if (A[P[0].x][P[0].y - i] == -1) {
                len3++;
            }
            else
                break;
        }
    }
    if (A[P[0].x][P[0].y + 1] == -1) { //下面可以走，計算長度
        for (int i = 1; i <= 9 - P[0].y; i++) {
            if (A[P[0].x][P[0].y + i] == -1) {
                len4++;
            }
            else
                break;
        }
    }
    int num1 = 0, x1 = 0, y1 = 0; //右邊
    int num2 = 0, x2 = 0, y2 = 0; //左邊
    int num3 = 0, x3 = 0, y3 = 0; //上面
    int num4 = 0, x4 = 0, y4 = 0; //下面
    //計算砲的十字路徑上是否有棋子
    for (int i = P[0].x + 1; i <= 8; i++) { //右邊
        if (A[i][P[0].y] != -1) {
            num1++;

```


位子

```
if (num1 == 2) { //如果有兩顆棋子紀錄其
```

```
    x1 = i;  
    y1 = P[0].y;
```

```
}
```

```
}
```

```
}
```

```
for (int i = P[0].x - 1; i >= 0; i--) { //左邊
```

```
    if (A[i][P[0].y] != -1) {
```

```
        num2++;
```

```
        if (num2 == 2) { //如果有兩顆棋子紀錄其
```

位子

```
            x2 = i;  
            y2 = P[0].y;
```

```
        }
```

```
    }
```

```
}
```

```
for (int i = P[0].y - 1; i >= 0; i--) { //上面
```

```
    if (A[P[0].x][i] != -1) {
```

```
        num3++;
```

```
        if (num3 == 2) { //如果有兩顆棋子紀錄其
```

位子

```
            x3 = P[0].x;  
            y3 = i;
```

```
        }
```

```
    }
```

```
}
```

```
for (int i = P[0].y + 1; i <= 9; i++) { //下面
```

```
    if (A[P[0].x][i] != -1) {
```

```
        num4++;
```

```
        if (num4 == 2) { //如果有兩顆棋子紀錄其
```

位子

```
            x4 = P[0].x;  
            y4 = i;
```

```
        }
```

```
    }
```

```
}
```

```
//若十字路徑上的第二顆棋是對方的則吃掉
```

```
if (num4 >= 2 && A[x4][y4] >= 100) { //吃下面
```

```
    x = x4;
```

```
    y = y4;
```

```
}
```

```
else if (num2 >= 2 && A[x2][y2] >= 100) { //吃左邊
```

```
    x = x2;
```

```
    y = y2;
```

```
}
```

```

else if (num1 >= 2 && A[x1][y1] >= 100) { //吃右邊
    x = x1;
    y = y1;
}
else if (num3 >= 2 && A[x3][y3] >= 100) { //吃上面
    x = x3;
    y = y3;
}
//沒有棋可以吃，在路徑最長的方向上亂數下棋
else if (len1 >= len2 && len1 >= len3 && len1 >= len4) { //往
左
    x = P[0].x - (rand() % (len1 - 1 + 1) + 1);
    y = P[0].y;
}
else if (len2 >= len1 && len2 >= len3 && len2 >= len4) //往右
{
    x = P[0].x + (rand() % (len2 - 1 + 1) + 1);
    y = P[0].y;
}
else if (len3 >= len2 && len3 >= len1 && len3 >= len4) { //往
上
    y = P[0].y - (rand() % (len3 - 1 + 1) + 1);
    x = P[0].x;
}
else if (len4 >= len2 && len4 >= len3 && len4 >= len1) { //往
下
    y = P[0].y + (rand() % (len4 - 1 + 1) + 1);
    x = P[0].x;
}
}
/*-----卒-----*/
else if (P[0].id == 11 || P[0].id == 12 || P[0].id == 13 || P[0].id == 14 ||
P[0].id == 15) {
    if (P[0].y <= 4 && A[P[0].x][P[0].y + 1] == -1 || P[0].y <= 4
&& A[P[0].x][P[0].y + 1] >= 100) //過河前
    {
        y = P[0].y + 1;
        x = P[0].x;
    }
    else if (P[0].y >= 5 && P[0].y <= 9) //過河後
    {
        //有棋子吃
        if (P[0].x + 1 <= 8 && A[P[0].x + 1][P[0].y] >= 100)
//往右吃
        {
            x = P[0].x + 1;
            y = P[0].y;

```

```

    }
    //往左吃
    else if (P[0].x - 1 >= 0 && A[P[0].x - 1][P[0].y] >= 100)
    {
        x = P[0].x - 1;
        y = P[0].y;
    }
    //往下吃
    else if (P[0].y + 1 <= 9 && A[P[0].x][P[0].y + 1] >=
100)
    {
        y = P[0].y + 1;
        x = P[0].x;
    }
    //沒有棋子吃，移動
    //往下
    else if (P[0].y + 1 <= 9 && A[P[0].x][P[0].y + 1] == -1)
    {
        y = P[0].y + 1;
        x = P[0].x;
    }
    //往右
    else if (P[0].x + 1 <= 8 && A[P[0].x + 1][P[0].y] == -1)
    {
        x = P[0].x + 1;
        y = P[0].y;
    }
    //往左
    else if (P[0].x - 1 >= 0 && A[P[0].x - 1][P[0].y] == -1)
    {
        x = P[0].x - 1;
        y = P[0].y;
    }
}
/*-----馬-----*/
else if (P[0].id == 8 || P[0].id == 7) {
    //有棋子吃
    if (A[P[0].x + 1][P[0].y + 2] >= 100 && A[P[0].x][P[0].y + 1]
== -1 && P[0].x + 1 <= 8 && P[0].y + 2 <= 9) //吃右下下
    {
        x = P[0].x + 1;
        y = P[0].y + 2;
    }
    else if (A[P[0].x + 2][P[0].y + 1] >= 100 && A[P[0].x +
1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y + 1 <= 9) { //吃右下
        x = P[0].x + 2;
        y = P[0].y + 1;
    }
}

```

```

        else if (A[P[0].x - 1][P[0].y + 2] >= 100 && A[P[0].x][P[0].y +
1] == -1 && P[0].x - 1 >= 0 && P[0].y + 2 <= 9) { //吃左下下
            x = P[0].x - 1;
            y = P[0].y + 2;
        }
        else if (A[P[0].x - 2][P[0].y + 1] >= 100 && A[P[0].x -
1][P[0].y] == -1 && P[0].x - 2 >= 0 && P[0].y + 1 <= 9) { //吃左下
            x = P[0].x - 2;
            y = P[0].y + 1;
        }
        else if (A[P[0].x + 2][P[0].y - 1] >= 100 && A[P[0].x +
1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y - 1 >= 0) { //吃右上
            x = P[0].x + 2;
            y = P[0].y - 1;
        }
        else if (A[P[0].x + 1][P[0].y - 2] >= 100 && A[P[0].x][P[0].y -
1] == -1 && P[0].x + 1 <= 8 && P[0].y - 2 >= 0) { //吃右上上
            x = P[0].x + 1;
            y = P[0].y - 2;
        }
        else if (A[P[0].x - 2][P[0].y - 1] >= 100 && A[P[0].x -
1][P[0].y] == -1 && P[0].x - 2 >= 0 && P[0].y - 1 >= 0) { //吃左上
            x = P[0].x - 2;
            y = P[0].y - 1;
        }
        else if (A[P[0].x - 1][P[0].y - 2] >= 100 && A[P[0].x][P[0].y -
1] == -1 && P[0].x - 1 >= 0 && P[0].y - 2 >= 0) { //吃左上上
            x = P[0].x - 1;
            y = P[0].y - 2;
        }
        //沒有棋子吃，移動
        else if (A[P[0].x + 1][P[0].y + 2] == -1 && A[P[0].x][P[0].y +
1] == -1 && P[0].x + 1 <= 8 && P[0].y + 2 <= 9) //右下下
        {
            x = P[0].x + 1;
            y = P[0].y + 2;
        }
        else if (A[P[0].x + 2][P[0].y + 1] == -1 && A[P[0].x +
1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y + 1 <= 9) { //右下
            x = P[0].x + 2;
            y = P[0].y + 1;
        }
        else if (A[P[0].x - 1][P[0].y + 2] == -1 && A[P[0].x][P[0].y +
1] == -1 && P[0].x - 1 >= 0 && P[0].y + 2 <= 9) { //左下下
            x = P[0].x - 1;
            y = P[0].y + 2;
        }
        else if (A[P[0].x - 2][P[0].y + 1] == -1 && A[P[0].x - 1][P[0].y]

```

```

== -1 && P[0].x - 2 >= 0 && P[0].y + 1 <= 9) { //左下
    x = P[0].x - 2;
    y = P[0].y + 1;
}
else if (A[P[0].x + 2][P[0].y - 1] == -1 && A[P[0].x +
1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y - 1 >= 0) { //右上
    x = P[0].x + 2;
    y = P[0].y - 1;
}
else if (A[P[0].x + 1][P[0].y - 2] == -1 && A[P[0].x][P[0].y - 1]
== -1 && P[0].x + 1 <= 8 && P[0].y - 2 >= 0) { //右上上
    x = P[0].x + 1;
    y = P[0].y - 2;
}
else if (A[P[0].x - 2][P[0].y - 1] == -1 && A[P[0].x - 1][P[0].y]
== -1 && P[0].x - 2 <= 8 && P[0].y - 1 <= 9) { //左上
    x = P[0].x - 2;
    y = P[0].y - 1;
}
else if (A[P[0].x - 1][P[0].y - 2] == -1 && A[P[0].x][P[0].y - 1]
== -1 && P[0].x - 1 <= 8 && P[0].y - 2 <= 9) { //左上上
    x = P[0].x - 1;
    y = P[0].y - 2;
}
}
/*-----象-----*/
else if (P[0].id == 3 || P[0].id == 4) {
    //有棋子吃
    if (A[P[0].x - 2][P[0].y + 2] >= 100 && A[P[0].x - 1][P[0].y +
1] == -1 && P[0].x - 2 >= 0 && P[0].y + 2 <= 4) { //吃左下
        x = P[0].x - 2;
        y = P[0].y + 2;
    }
    else if (A[P[0].x - 2][P[0].y - 2] >= 100 && A[P[0].x -
1][P[0].y - 1] == -1 && P[0].x - 2 >= 8 && P[0].y - 2 >= 0) { //吃左上
        x = P[0].x - 2;
        y = P[0].y - 2;
    }
    else if (A[P[0].x + 2][P[0].y - 2] >= 100 && A[P[0].x +
1][P[0].y - 1] == -1 && P[0].x + 2 <= 8 && P[0].y - 2 >= 0) { //吃右上
        x = P[0].x + 2;
        y = P[0].y - 2;
    }
    else if (A[P[0].x + 2][P[0].y + 2] >= 100 && A[P[0].x +
1][P[0].y + 1] == -1 && P[0].x + 2 <= 8 && P[0].y + 2 <= 4) { //吃右下
        x = P[0].x + 2;
        y = P[0].y + 2;
    }
}

```

```

//沒有棋子吃，移動
else if (A[P[0].x - 2][P[0].y + 2] == -1 && A[P[0].x - 1][P[0].y
+ 1] == -1 && P[0].x - 2 >= 0 && P[0].y + 2 <= 4) { //左下
    x = P[0].x - 2;
    y = P[0].y + 2;
}
else if (A[P[0].x - 2][P[0].y - 2] == -1 && A[P[0].x - 1][P[0].y -
1] == -1 && P[0].x - 2 >= 0 && P[0].y - 2 >= 0) { //左上
    x = P[0].x - 2;
    y = P[0].y - 2;
}
else if (A[P[0].x + 2][P[0].y - 2] == -1 && A[P[0].x + 1][P[0].y
- 1] == -1 && P[0].x + 2 <= 8 && P[0].y - 2 >= 0) { //右上
    x = P[0].x + 2;
    y = P[0].y - 2;
}
else if (A[P[0].x + 2][P[0].y + 2] == -1 && A[P[0].x +
1][P[0].y + 1] == -1 && P[0].x + 2 <= 8 && P[0].y + 2 <= 4) { //右下
    x = P[0].x + 2;
    y = P[0].y + 2;
}
}
/*-----士-----*/
else if (P[0].id == 1 || P[0].id == 2) {
//有棋子吃
if (A[P[0].x - 1][P[0].y + 1] >= 100 && P[0].x - 1 >= 3 &&
P[0].y + 1 <= 2) //吃左斜下
{
    x = P[0].x - 1;
    y = P[0].y + 1;
}
else if (A[P[0].x + 1][P[0].y + 1] >= 100 && P[0].x + 1 <= 5
&& P[0].y + 1 <= 2) //吃右斜下
{
    x = P[0].x + 1;
    y = P[0].y + 1;
}
else if (A[P[0].x - 1][P[0].y - 1] >= 100 && P[0].x - 1 <= 5
&& P[0].y - 1 >= 0) //吃右斜上
{
    x = P[0].x - 1;
    y = P[0].y - 1;
}
else if (A[P[0].x + 1][P[0].y - 1] >= 100 && P[0].x + 1 >= 3
&& P[0].y - 1 >= 0) //吃左斜上
{
    x = P[0].x + 1;
    y = P[0].y - 1;
}
}

```

```

    }
    //沒有棋子吃，移動
    else if (A[P[0].x - 1][P[0].y + 1] == -1 && P[0].x - 1 >= 3 &&
P[0].y + 1 <= 2) //左斜下
    {
        x = P[0].x - 1;
        y = P[0].y + 1;
    }
    else if (A[P[0].x + 1][P[0].y + 1] == -1 && P[0].x + 1 <= 5 &&
P[0].y + 1 <= 2) //右斜下
    {
        x = P[0].x + 1;
        y = P[0].y + 1;
    }
    else if (A[P[0].x + 1][P[0].y - 1] == -1 && P[0].x + 1 <= 5 &&
P[0].y - 1 >= 0) //右斜上
    {
        x = P[0].x + 1;
        y = P[0].y - 1;
    }
    else if (A[P[0].x - 1][P[0].y - 1] == -1 && P[0].x - 1 >= 3 &&
P[0].y - 1 >= 0) //左斜上
    {
        x = P[0].x - 1;
        y = P[0].y - 1;
    }
}
/*-----將-----*/
else if (P[0].id == 0) {
    //有棋子吃
    if (A[P[0].x][P[0].y + 1] >= 100 && P[0].y + 1 <= 2) //往
下吃
    {
        x = P[0].x;
        y = P[0].y + 1;
    }

    else if (A[P[0].x + 1][P[0].y] >= 100 && P[0].x + 1 <= 5) //往
右吃
    {
        x = P[0].x + 1;
        y = P[0].y;
    }

    else if (A[P[0].x - 1][P[0].y] >= 100 && P[0].x - 1 <= 5) //往
左吃
    {
        x = P[0].x - 1;
        y = P[0].y;
    }
}

```

```

    }
    else if (A[P[0].x][P[0].y - 1] >= 100 && P[0].y - 1 >= 0)    //往
上吃
    {
        x = P[0].x;
        y = P[0].y - 1;
    }
    //沒有棋子吃，移動
    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].y + 1 <= 2)    //往
下走
    {
        x = P[0].x;
        y = P[0].y + 1;
    }

    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].x + 1 <= 5)    //往
右走
    {
        x = P[0].x + 1;
        y = P[0].y;
    }

    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].x - 1 <= 5)    //往
左走
    {
        x = P[0].x - 1;
        y = P[0].y;
    }

    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].y - 1 >= 0)    //往
上走
    {
        x = P[0].x;
        y = P[0].y - 1;
    }
    }
}

/*-----紅棋-----*/
else if (order == 1)
{
    //帥的防守
    int king = 0;
    for (int i = 0; i < 16; i++)
    {
        if (P[i].id == 100)
        {
            king = i;

```



```

        break;
    }
}
if (A[P[king].x + 1][P[king].y] < 100 && A[P[king].x + 1][P[king].y] != -1
&& P[king].x + 1 <= 5)    //帥的正右方有東西吃
{
    x = P[king].x + 1;
    y = P[king].y;
    P[0].id = 100;
}
else if (A[P[king].x - 1][P[king].y] < 100 && A[P[king].x - 1][P[king].y] != -1
&& P[king].x - 1 >= 3)    //帥的正左方有東西吃
{
    x = P[king].x - 1;
    y = P[king].y;
    P[0].id = 100;
}
else if (A[P[king].x][P[king].y + 1] < 100 && A[P[king].x][P[king].y + 1] != -1
&& P[king].y + 1 <= 9) //帥的正下方有東西吃
{
    x = P[king].x;
    y = P[king].y + 1;
    P[0].id = 100;
}
else if (A[P[king].x][P[king].y - 1] < 100 && A[P[king].x][P[king].y - 1] != -1
&& P[king].y - 1 >= 7)    //帥的正上方有東西吃
{
    x = P[king].x;
    y = P[king].y - 1;
    P[0].id = 100;
}
//若帥周圍沒有對手棋
else
{
    for (int i = 0; i < 16; i++) {
        for (int j = 0; j < 16; j++) {
            if (P[i].score > P[j].score) {
                int tmp = P[i].score, tmp2 = P[i].id, tmp3 =
P[i].x, tmp4 = P[i].y;
                P[i].score = P[j].score;
                P[i].id = P[j].id;
                P[i].x = P[j].x;
                P[i].y = P[j].y;
                P[j].score = tmp;
                P[j].id = tmp2;
                P[j].x = tmp3;
                P[j].y = tmp4;
            }
        }
    }
}

```

```

    }
    int len1 = 0; //左邊
    int len2 = 0; //右邊
    int len3 = 0; //後面
    int len4 = 0; //前面
    /*-----車-----*/
    if (P[0].id == 106 || P[0].id == 105) {
        if (A[P[0].x - 1][P[0].y] == -1) { //若左邊可以走，計算長度
            for (int i = 1; i <= P[0].x; i++) {
                if (A[P[0].x - i][P[0].y] == -1) {
                    len1++;
                }
                else
                    break;
            }
        }
        if (A[P[0].x + 1][P[0].y] == -1) { //若右邊可以走，計算長度
            for (int i = 1; i <= 8 - P[0].x; i++) {
                if (A[P[0].x + i][P[0].y] == -1) {
                    len2++;
                }
                else
                    break;
            }
        }
        if (A[P[0].x][P[0].y - 1] == -1) { //若上面可以走，計算長度
            for (int i = 1; i <= P[0].y; i++) {
                if (A[P[0].x][P[0].y - i] == -1) {
                    len3++;
                }
                else
                    break;
            }
        }
        if (A[P[0].x][P[0].y + 1] == -1) { //若下面可以走，計算長度
            for (int i = 1; i <= 9 - P[0].y; i++) {
                if (A[P[0].x][P[0].y + i] == -1) {
                    len4++;
                }
                else
                    break;
            }
        }
        //判斷車能否吃將
        if (A[P[0].x][P[0].y + len4 + 1] == 0 && P[0].y + len4 + 1 <=
9) { //往下吃將
            x = P[0].x;
            y = P[0].y + len4 + 1;

```

```

    }
    else if (A[P[0].x - len1 - 1][P[0].y] == 0 && P[0].x - len1 - 1
    >= 0) { //往左吃將
        x = P[0].x - len1 - 1;
        y = P[0].y;
    }
    else if (A[P[0].x + len2 + 1][P[0].y] == 0 && P[0].x + len2 + 1
    <= 8) { //往右吃將
        x = P[0].x + len2 + 1;
        y = P[0].y;
    }
    else if (A[P[0].x][P[0].y - len3 - 1] == 0 && P[0].y - len3 - 1
    >= 0) { //往上吃將
        x = P[0].x;
        y = P[0].y - len3 - 1;
    }
    //若不是將，但為對手的棋子
    else if (A[P[0].x][P[0].y + len4 + 1] < 100 && P[0].y + len4 + 1
    <= 9) { //往下吃
        x = P[0].x;
        y = P[0].y + len4 + 1;
    }
    else if (A[P[0].x - len1 - 1][P[0].y] < 100 && P[0].x - len1 - 1
    >= 0) { //往左吃
        x = P[0].x - len1 - 1;
        y = P[0].y;
    }
    else if (A[P[0].x + len2 + 1][P[0].y] < 100 && P[0].x + len2 + 1
    <= 8) { //往右吃
        x = P[0].x + len2 + 1;
        y = P[0].y;
    }
    else if (A[P[0].x][P[0].y - len3 - 1] < 100 && P[0].y - len3 - 1
    >= 0) { //往上吃
        x = P[0].x;
        y = P[0].y - len3 - 1;
    }
    //都沒有得吃，在路徑最長的方向上亂數下棋
    else if (len1 >= len2 && len1 >= len3 && len1 >= len4) { //往
    左
        x = P[0].x - (rand() % (len1 - 1) + 1);
        y = P[0].y;
    }
    else if (len2 >= len1 && len2 >= len3 && len2 >= len4) //往右
    {
        x = P[0].x + (rand() % (len2 - 1) + 1);
        y = P[0].y;
    }
}

```

上

下

```
else if (len3 >= len2 && len3 >= len1 && len3 >= len4) { //往
    y = P[0].y - (rand() % (len3 - 1) + 1);
    x = P[0].x;
}
else if (len4 >= len2 && len4 >= len3 && len4 >= len1) { //往
    y = P[0].y + (rand() % (len4 - 1) + 1);
    x = P[0].x;
}
}
/*-----炮-----*/
else if (P[0].id == 109 || P[0].id == 110) {
    if (A[P[0].x - 1][P[0].y] == -1) { //左邊可以走
        for (int i = 1; i <= P[0].x; i++) {
            if (A[P[0].x - i][P[0].y] == -1) {
                len1++;
            }
            else
                break;
        }
    }
    if (A[P[0].x + 1][P[0].y] == -1) { //右邊可以走
        for (int i = 1; i <= 8 - P[0].x; i++) {
            if (A[P[0].x + i][P[0].y] == -1) {
                len2++;
            }
            else
                break;
        }
    }
    if (A[P[0].x][P[0].y - 1] == -1) { //上面可以走
        for (int i = 1; i <= P[0].y; i++) {
            if (A[P[0].x][P[0].y - i] == -1) {
                len3++;
            }
            else
                break;
        }
    }
    if (A[P[0].x][P[0].y + 1] == -1) { //下面可以走
        for (int i = 1; i <= 9 - P[0].y; i++) {
            if (A[P[0].x][P[0].y + i] == -1) {
                len4++;
            }
            else
                break;
        }
    }
}
```

位子

```
}
int num1 = 0, x1 = 0, y1 = 0; //右邊
int num2 = 0, x2 = 0, y2 = 0; //左邊
int num3 = 0, x3 = 0, y3 = 0; //上面
int num4 = 0, x4 = 0, y4 = 0; //下面
//計算炮的十字路徑上是否有棋子
for (int i = P[0].x + 1; i <= 8; i++) { //右邊
    if (A[i][P[0].y] != -1) {
        num1++;
        if (num1 == 2) { //如果有兩顆棋子紀錄其
            x1 = i;
            y1 = P[0].y;
        }
    }
}
```

位子

```
}
for (int i = P[0].x - 1; i >= 0; i--) { //左邊
    if (A[i][P[0].y] != -1) {
        num2++;
        if (num2 == 2) { //如果有兩顆棋子紀錄其
            x2 = i;
            y2 = P[0].y;
        }
    }
}
```

位子

```
}
for (int i = P[0].y - 1; i >= 0; i--) { //上面
    if (A[P[0].x][i] != -1) {
        num3++;
        if (num3 == 2) { //如果有兩顆棋子紀錄其
            x3 = P[0].x;
            y3 = i;
        }
    }
}
```

位子

```
}
for (int i = P[0].y + 1; i <= 9; i++) { //下面
    if (A[P[0].x][i] != -1) {
        num4++;
        if (num4 == 2) { //如果有兩顆棋子紀錄其
            x4 = P[0].x;
            y4 = i;
        }
    }
}
```

//若十字路徑上的第二顆棋是對方的則吃掉

```

if (num3 >= 2 && A[x3][y3] < 100) { //吃上面
    x = x3;
    y = y3;
}
else if (num4 >= 2 && A[x4][y4] < 100) { //吃下面
    x = x4;
    y = y4;
}
else if (num2 >= 2 && A[x2][y2] < 100) { //吃左邊
    x = x2;
    y = y2;
}
else if (num1 >= 2 && A[x1][y1] < 100) { //吃右邊
    x = x1;
    y = y1;
}
//沒有棋可以吃，路徑上亂數下棋
else if (len1 >= len2 && len1 >= len3 && len1 >= len4) { //往
左
    x = P[0].x - (rand() % (len1 - 1 + 1) + 1);
    y = P[0].y;
}
else if (len2 >= len1 && len2 >= len3 && len2 >= len4) //往右
{
    x = P[0].x + (rand() % (len2 - 1 + 1) + 1);
    y = P[0].y;
}
else if (len3 >= len2 && len3 >= len1 && len3 >= len4) { //往
上
    y = P[0].y - (rand() % (len3 - 1 + 1) + 1);
    x = P[0].x;
}
else if (len4 >= len2 && len4 >= len3 && len4 >= len1) { //往
下
    y = P[0].y + (rand() % (len4 - 1 + 1) + 1);
    x = P[0].x;
}
}
/*-----兵-----*/
else if (P[0].id == 111 || P[0].id == 112 || P[0].id == 113 || P[0].id ==
114 || P[0].id == 115) {
    if (P[0].y >= 5 && A[P[0].x][P[0].y - 1] == -1 || P[0].y >= 5
&& A[P[0].x][P[0].y - 1] < 100) //過河前
    {
        y = P[0].y - 1;
        x = P[0].x;
    }
    else if (P[0].y >= 0 && P[0].y <= 4) //過河後

```

```

        {
            //有棋子吃
            if (P[0].x + 1 <= 8 && A[P[0].x + 1][P[0].y] < 100 &&
A[P[0].x + 1][P[0].y] != -1) //往右吃
            {
                x = P[0].x + 1;
                y = P[0].y;
            }
            else if (P[0].x - 1 >= 0 && A[P[0].x - 1][P[0].y] < 100
&& A[P[0].x - 1][P[0].y] != -1) //往左吃
            {
                x = P[0].x - 1;
                y = P[0].y;
            }
            else if (P[0].y - 1 >= 0 && A[P[0].x][P[0].y - 1] < 100
&& A[P[0].x][P[0].y - 1] != -1) //往上吃
            {
                y = P[0].y - 1;
                x = P[0].x;
            }
            //沒有棋子吃，移動
            else if (P[0].y - 1 >= 0 && A[P[0].x][P[0].y - 1] == -1)
//往上
            {
                y = P[0].y - 1;
                x = P[0].x;
            }
            else if (P[0].x + 1 <= 8 && A[P[0].x + 1][P[0].y] == -1)
//往右
            {
                x = P[0].x + 1;
                y = P[0].y;
            }
            else if (P[0].x - 1 >= 0 && A[P[0].x - 1][P[0].y] == -1)
//往左
            {
                x = P[0].x - 1;
                y = P[0].y;
            }
        }
    }
    /*-----馬-----*/
    else if (P[0].id == 108 || P[0].id == 107) {
        //吃棋
        if (A[P[0].x + 1][P[0].y + 2] != -1 && A[P[0].x + 1][P[0].y + 2]
< 100 && A[P[0].x][P[0].y + 1] == -1 && P[0].x + 1 <= 8 && P[0].y + 2 <= 9) //吃右下下
        {
            x = P[0].x + 1;

```

```

        y = P[0].y + 2;
    }
    else if (A[P[0].x + 2][P[0].y + 1] != -1 && A[P[0].x + 2][P[0].y
+ 1] < 100 && A[P[0].x + 1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y + 1 <= 9) { //吃右
下
        x = P[0].x + 2;
        y = P[0].y + 1;
    }
    else if (A[P[0].x - 1][P[0].y + 2] != -1 && A[P[0].x - 1][P[0].y
+ 2] < 100 && A[P[0].x][P[0].y + 1] == -1 && P[0].x - 1 >= 0 && P[0].y + 2 <= 9) { //吃左
下下
        x = P[0].x - 1;
        y = P[0].y + 2;
    }
    else if (A[P[0].x - 2][P[0].y + 1] != -1 && A[P[0].x - 2][P[0].y
+ 1] < 100 && A[P[0].x - 1][P[0].y] == -1 && P[0].x - 2 >= 0 && P[0].y + 1 <= 9) { //吃左
下
        x = P[0].x - 2;
        y = P[0].y + 1;
    }
    else if (A[P[0].x + 2][P[0].y - 1] != -1 && A[P[0].x + 2][P[0].y
- 1] < 100 && A[P[0].x + 1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y - 1 >= 0) { //吃右
上
        x = P[0].x + 2;
        y = P[0].y - 1;
    }
    else if (A[P[0].x + 1][P[0].y - 2] != -1 && A[P[0].x + 1][P[0].y
- 2] < 100 && A[P[0].x][P[0].y - 1] == -1 && P[0].x + 1 <= 8 && P[0].y - 2 >= 0) { //吃右
上上
        x = P[0].x + 1;
        y = P[0].y - 2;
    }
    else if (A[P[0].x - 2][P[0].y - 1] != -1 && A[P[0].x - 2][P[0].y -
1] < 100 && A[P[0].x - 1][P[0].y] == -1 && P[0].x - 2 >= 0 && P[0].y - 1 >= 0) { //吃左上
        x = P[0].x - 2;
        y = P[0].y - 1;
    }
    else if (A[P[0].x - 1][P[0].y - 2] != -1 && A[P[0].x - 1][P[0].y -
2] < 100 && A[P[0].x][P[0].y - 1] == -1 && P[0].x - 1 >= 0 && P[0].y - 2 >= 0) { //吃左上
上
        x = P[0].x - 1;
        y = P[0].y - 2;
    }
    //沒有棋吃，移動
    else if (A[P[0].x + 1][P[0].y + 2] == -1 && A[P[0].x][P[0].y +
1] == -1 && P[0].x + 1 <= 8 && P[0].y + 2 <= 9) //右下下
    {
        x = P[0].x + 1;

```



```

        y = P[0].y + 2;
    }
    else if (A[P[0].x + 2][P[0].y + 1] == -1 && A[P[0].x +
1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y + 1 <= 9) { //右下
        x = P[0].x + 2;
        y = P[0].y + 1;
    }
    else if (A[P[0].x - 1][P[0].y + 2] == -1 && A[P[0].x][P[0].y +
1] == -1 && P[0].x - 1 >= 0 && P[0].y + 2 <= 9) { //左下下
        x = P[0].x - 1;
        y = P[0].y + 2;
    }
    else if (A[P[0].x - 2][P[0].y + 1] == -1 && A[P[0].x - 1][P[0].y]
== -1 && P[0].x - 2 >= 0 && P[0].y + 1 <= 9) { //左下
        x = P[0].x - 2;
        y = P[0].y + 1;
    }
    else if (A[P[0].x + 2][P[0].y - 1] == -1 && A[P[0].x +
1][P[0].y] == -1 && P[0].x + 2 <= 8 && P[0].y - 1 >= 0) { //右上
        x = P[0].x + 2;
        y = P[0].y - 1;
    }
    else if (A[P[0].x + 1][P[0].y - 2] == -1 && A[P[0].x][P[0].y - 1]
== -1 && P[0].x + 1 <= 8 && P[0].y - 2 >= 0) { //右上上
        x = P[0].x + 1;
        y = P[0].y - 2;
    }
    else if (A[P[0].x - 2][P[0].y - 1] == -1 && A[P[0].x - 1][P[0].y]
== -1 && P[0].x - 2 <= 8 && P[0].y - 1 <= 9) { //左上
        x = P[0].x - 2;
        y = P[0].y - 1;
    }
    else if (A[P[0].x - 1][P[0].y - 2] == -1 && A[P[0].x][P[0].y - 1]
== -1 && P[0].x - 1 <= 8 && P[0].y - 2 <= 9) { //左上上
        x = P[0].x - 1;
        y = P[0].y - 2;
    }
}
/*-----相-----*/
else if (P[0].id == 103 || P[0].id == 104) {
    //吃棋
    if (A[P[0].x - 2][P[0].y + 2] != -1 && A[P[0].x - 2][P[0].y + 2]
< 100 && A[P[0].x - 1][P[0].y + 1] == -1 && P[0].x - 2 >= 0 && P[0].y + 2 <= 9) { //吃左
下
        x = P[0].x - 2;
        y = P[0].y + 2;
    }
    else if (A[P[0].x - 2][P[0].y - 2] != -1 && A[P[0].x - 2][P[0].y -

```

2] < 100 && A[P[0].x - 1][P[0].y - 1] == -1 && P[0].x - 2 >= 8 && P[0].y - 2 >= 5) { //吃左上

```

    x = P[0].x - 2;
    y = P[0].y - 2;

```

```

}

```

else if (A[P[0].x + 2][P[0].y - 2] != -1 && A[P[0].x + 2][P[0].y - 2] < 100 && A[P[0].x + 1][P[0].y - 1] == -1 && P[0].x + 2 <= 8 && P[0].y - 2 >= 5) { //吃右上

```

    x = P[0].x + 2;
    y = P[0].y - 2;

```

```

}

```

else if (A[P[0].x + 2][P[0].y + 2] != -1 && A[P[0].x + 2][P[0].y + 2] < 100 && A[P[0].x + 1][P[0].y + 1] == -1 && P[0].x + 2 <= 8 && P[0].y + 2 <= 9) { //吃右下

```

    x = P[0].x + 2;
    y = P[0].y + 2;

```

```

}

```

//沒有棋吃，移動

else if (A[P[0].x - 2][P[0].y + 2] == -1 && A[P[0].x - 1][P[0].y + 1] == -1 && P[0].x - 2 >= 0 && P[0].y + 2 <= 9) { //左下

```

    x = P[0].x - 2;
    y = P[0].y + 2;

```

```

}

```

else if (A[P[0].x - 2][P[0].y - 2] == -1 && A[P[0].x - 1][P[0].y - 1] == -1 && P[0].x - 2 >= 0 && P[0].y - 2 >= 5) { //左上

```

    x = P[0].x - 2;
    y = P[0].y - 2;

```

```

}

```

else if (A[P[0].x + 2][P[0].y - 2] == -1 && A[P[0].x + 1][P[0].y - 1] == -1 && P[0].x + 2 <= 8 && P[0].y - 2 >= 5) { //右上

```

    x = P[0].x + 2;
    y = P[0].y - 2;

```

```

}

```

else if (A[P[0].x + 2][P[0].y + 2] == -1 && A[P[0].x + 1][P[0].y + 1] == -1 && P[0].x + 2 <= 8 && P[0].y + 2 <= 9) { //右下

```

    x = P[0].x + 2;
    y = P[0].y + 2;

```

```

}

```

```

}

```

/*-----士-----*/

else if (P[0].id == 101 || P[0].id == 102) {

//吃棋

if (A[P[0].x - 1][P[0].y + 1] != -1 && A[P[0].x - 1][P[0].y + 1] < 100 && P[0].x - 1 >= 3 && P[0].y + 1 <= 9) //吃左斜下

```

{

```

```

    x = P[0].x - 1;
    y = P[0].y + 1;

```

```

}

```

```

        else if (A[P[0].x + 1][P[0].y + 1] != -1 && A[P[0].x + 1][P[0].y
+ 1] < 100 && P[0].x + 1 <= 5 && P[0].y + 1 <= 9)//吃右斜下
        {
            x = P[0].x + 1;
            y = P[0].y + 1;
        }
        else if (A[P[0].x - 1][P[0].y + 1] != -1 && A[P[0].x - 1][P[0].y
+ 1] < 100 && P[0].x + 1 <= 5 && P[0].y - 1 >= 7) //吃右斜上
        {
            x = P[0].x + 1;
            y = P[0].y - 1;
        }
        else if (A[P[0].x - 1][P[0].y + 1] != -1 && A[P[0].x - 1][P[0].y
+ 1] < 100 && P[0].x - 1 >= 3 && P[0].y - 1 >= 7) //吃左斜上
        {
            x = P[0].x - 1;
            y = P[0].y - 1;
        }
        //沒有棋吃，移動
        else if (A[P[0].x - 1][P[0].y + 1] == -1 && P[0].x - 1 >= 3 &&
P[0].y + 1 <= 9) //左斜下
        {
            x = P[0].x - 1;
            y = P[0].y + 1;
        }
        else if (A[P[0].x + 1][P[0].y + 1] == -1 && P[0].x + 1 <= 5 &&
P[0].y + 1 <= 9) //右斜下
        {
            x = P[0].x + 1;
            y = P[0].y + 1;
        }
        else if (A[P[0].x + 1][P[0].y - 1] == -1 && P[0].x + 1 <= 5 &&
P[0].y - 1 >= 7) //右斜上
        {
            x = P[0].x + 1;
            y = P[0].y - 1;
        }
        else if (A[P[0].x - 1][P[0].y - 1] == -1 && P[0].x - 1 >= 3 &&
P[0].y - 1 >= 7) //左斜上
        {
            x = P[0].x - 1;
            y = P[0].y - 1;
        }
    }
    /*-----帥-----*/
    else if (P[0].id == 100) {
        //吃棋
        if (A[P[0].x][P[0].y + 1] != -1 && A[P[0].x][P[0].y + 1] < 100

```

```

    && P[0].y + 1 <= 9) //往下吃
    {
        x = P[0].x;
        y = P[0].y + 1;
    }

    else if (A[P[0].x + 1][P[0].y] != -1 && A[P[0].x + 1][P[0].y] <
100 && P[0].x + 1 <= 5) //往右吃
    {
        x = P[0].x + 1;
        y = P[0].y;
    }

    else if (A[P[0].x - 1][P[0].y] != -1 && A[P[0].x - 1][P[0].y] <
100 && P[0].x - 1 <= 5) //往左吃
    {
        x = P[0].x - 1;
        y = P[0].y;
    }

    else if (A[P[0].x][P[0].y - 1] != -1 && A[P[0].x][P[0].y - 1] <
100 && P[0].y - 1 >= 7) //往上吃
    {
        x = P[0].x;
        y = P[0].y - 1;
    }

    //沒有棋吃，移動
    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].y + 1 <= 9) //往
下走
    {
        x = P[0].x;
        y = P[0].y + 1;
    }

    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].x + 1 <= 5) //往
右走
    {
        x = P[0].x + 1;
        y = P[0].y;
    }

    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].x - 1 <= 5) //往
左走
    {
        x = P[0].x - 1;
        y = P[0].y;
    }

    else if (A[P[0].x][P[0].y + 1] == -1 && P[0].y - 1 >= 7) //往
上走
    {
        x = P[0].x;
        y = P[0].y - 1;
    }

```

```

        }
    }
}

}
//-- write the decision ---

ofstream file;

file.open("play.txt");
file << order << " " << P[0].id << " " << x << " " << y << endl;

file.close();
return 0;
}

```