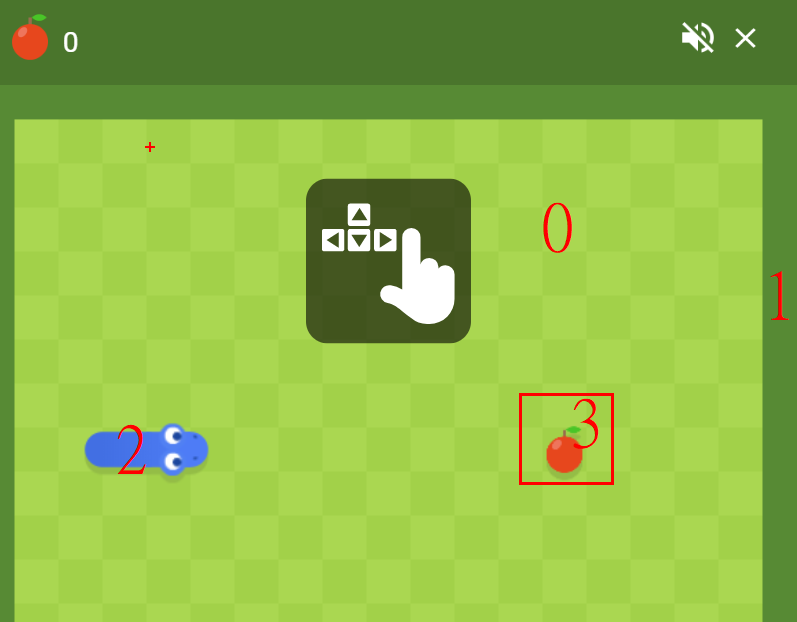
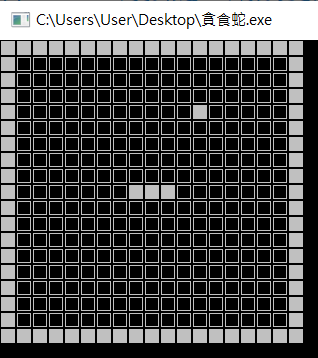
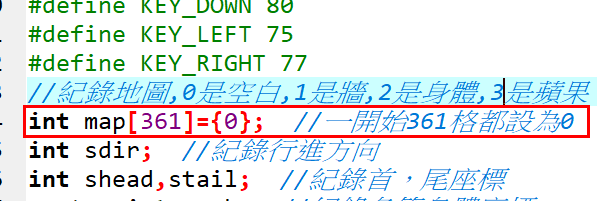


遊戲裡要記錄四種不同的東西，分別是空白格0，牆面1，蛇2，蘋果3。(你可以用別的代號，只要不重複即可)



在C++裡面受限於繪圖，我們只用黑白方塊來呈現，空格用白方塊，其餘用黑方塊。

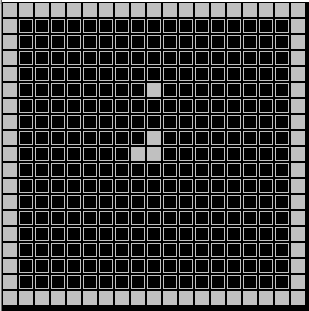


19\*19方陣，可以宣告361格的陣列來存取。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

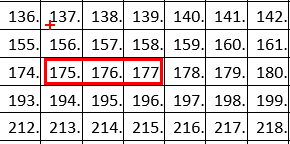
|  |
| --- |
| #include <conio.h>  #include <iostream>  #include <vector>  #include <algorithm>  #include <time.h>  #include <windows.h>  using namespace std;  //定義方向鍵  #define KEY\_UP 72  #define KEY\_DOWN 80  #define KEY\_LEFT 75  #define KEY\_RIGHT 77  //紀錄地圖,0是空白,1是牆,2是身體,3是蘋果  int map[361]={0}; //一開始361格都設為0  int sdir; //紀錄行進方向  int shead,stail; //紀錄首，尾座標  vector<int> snk; //紀錄各節身體座標  int apple; //紀錄蘋果座標  //第一題，地圖繪製，0用白色方塊顯示，其餘用黑色方塊  void draw\_map() {  system("cls"); //cls means clear screen，藉由重繪來表現貪食蛇的移動  for(int i=0;i<361;i++) {  if (map[i]==0)  cout<<"□";  else  cout<<"■";  //19\*19，每19塊要換行  if (i%19==18)  cout<<endl;  }  }  //第二題，設定邊界與加入蛇，設定一開始位置 175 176 177，蛇的代號是2  void init\_map() {  //設定邊界 ，邊界的代號是1  for(int i=0;i<=18;i++)//上排  map[i]=1;  for(int i=342;i<=360;i++)//下排  map[i]=1;  for(int i=0;i<=342;i+=19)//左排  map[i]=1;  for(int i=18;i<=360;i+=19)//右排  map[i]=1;  // 設定一開始位置 175 176 177，蛇的代號是2  map[175]=2;  map[176]=2;  map[177]=2;  // 記錄各節身體 175, 176, 177， 每次push back就會往右邊加入一項，也就是最新的蛇首座標  snk.push\_back(175);  snk.push\_back(176);  snk.push\_back(177);  // 設定蛇首座標  shead=177;  // 設定初始方向，向右  sdir=1;  // 呼叫第一題做繪圖，其實也可以把第一題全部程式碼剪貼到這，只是這樣寫比較簡潔。  draw\_map();  }  //第三題，加入蘋果 ，蘋果的代碼是3  void new\_apple() {  int x;  while(1) { //無窮迴圈  x = rand() % 361; //亂數抓一個位置  if (map[x]==0) //空白的位置才可以，要排除牆壁，蛇  break; //break可以跳出迴圈，也就是找到第一個空白位置，就可以離開while迴圈  }  apple=x;  map[apple]=3; //設定蘋果座標與代碼  }  int main(void)  {  srand( (int)time(NULL));//重置亂數表  int k; //紀錄你按到哪一個按鍵  init\_map();  new\_apple();  draw\_map();  do {  // 有按鍵，則改變方向  k = \_getch(); //\_getch means get a char 鍵盤按鍵，a b c d...也就是一個字元 ，但是方向鍵比較特殊，會送出兩個字元，其中第一個是特殊字元224  if (k == 224) { // 比較第一個字元，若是224，表示是方向鍵  k = \_getch(); // 讀取第二個字元，藉由判斷是哪個方向鍵，方向鍵我已經幫你定義在最上面 ，所以你可以直接用  }  } while (k != 27); // 按鍵盤左上方的ESC，可以結束程式  return 0;  } |

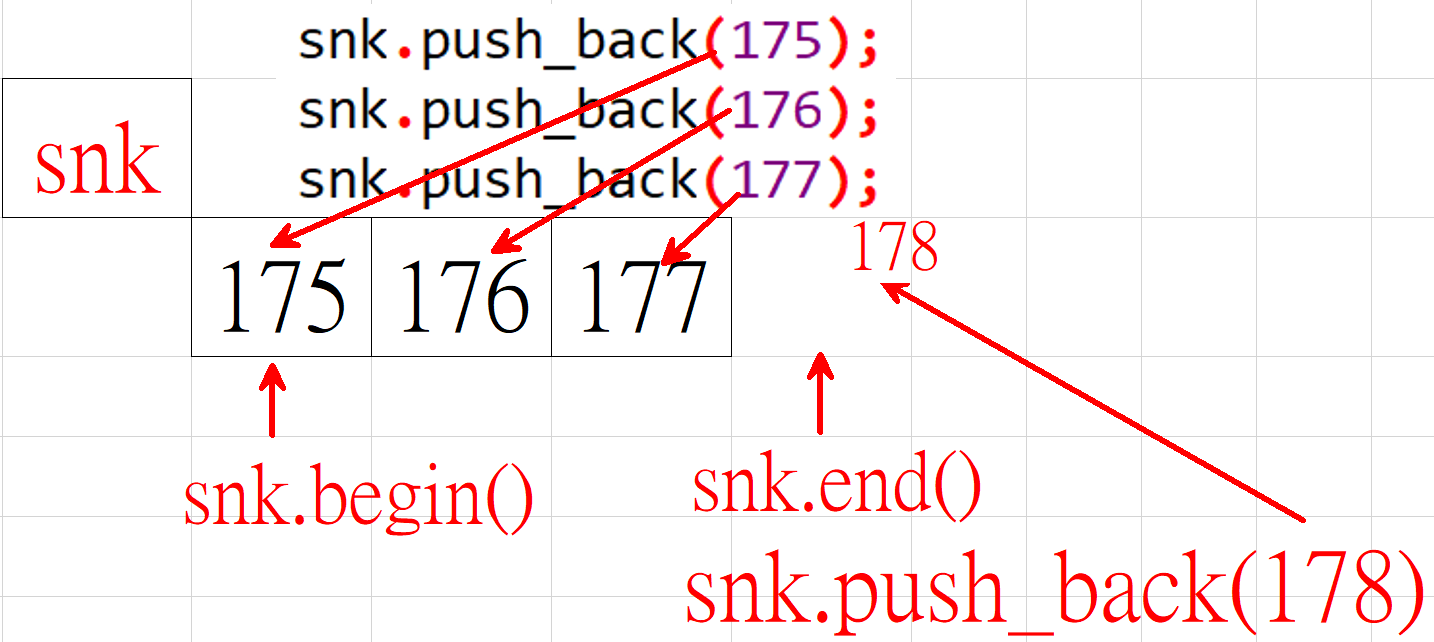
第一題，可以使用鍵盤上下左右鍵控制貪食蛇方向，吃蘋果會長一格。

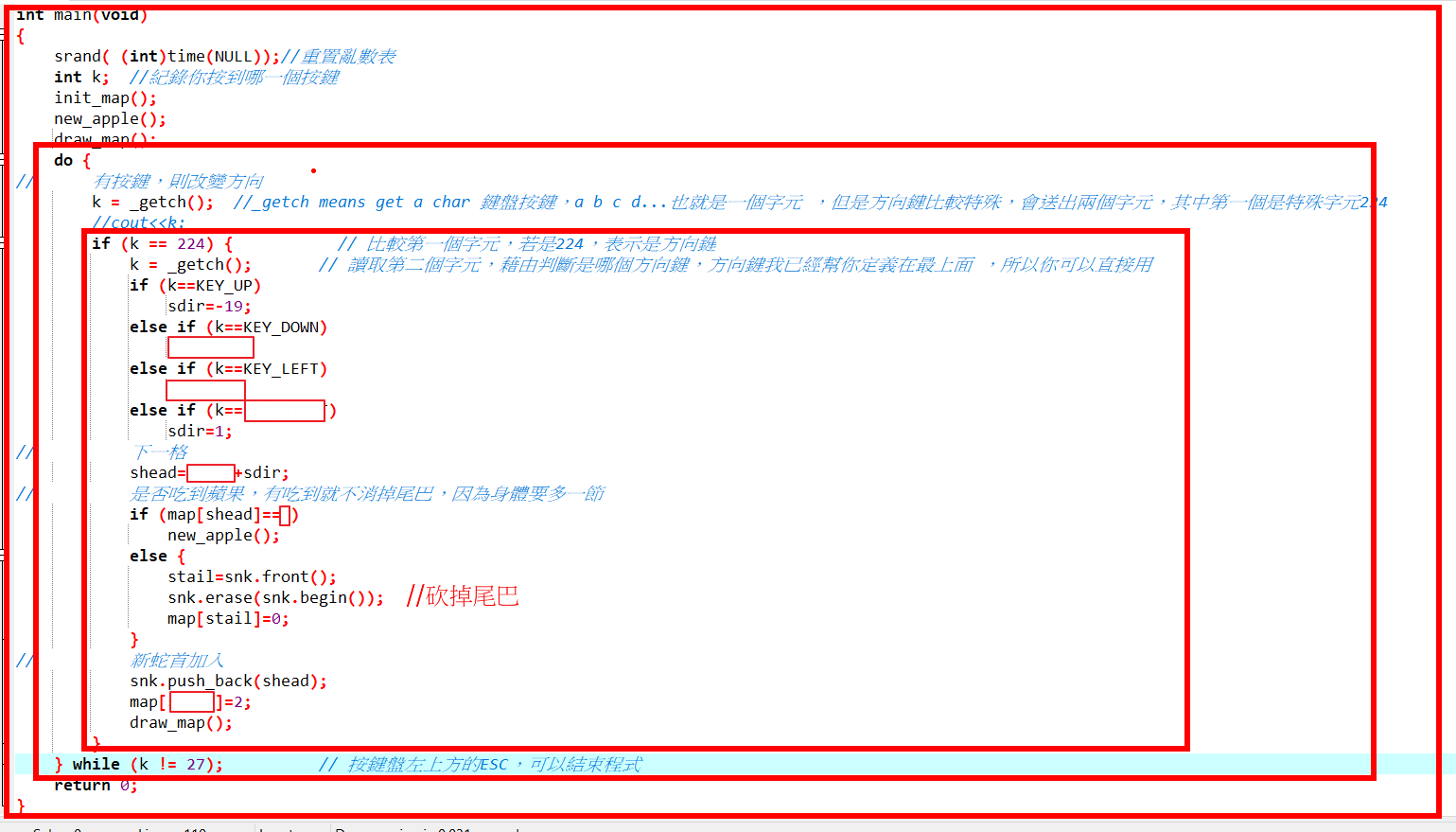


Vector 是 C++ 中一個非常好用的「容器」，是加強版的陣列，對自己的一些基本資訊提供成員函式來直接存取。

由於蛇是會移動的，比如現在有三節，我們要把座標175 176 177記錄下來。則snk現在有三個元素

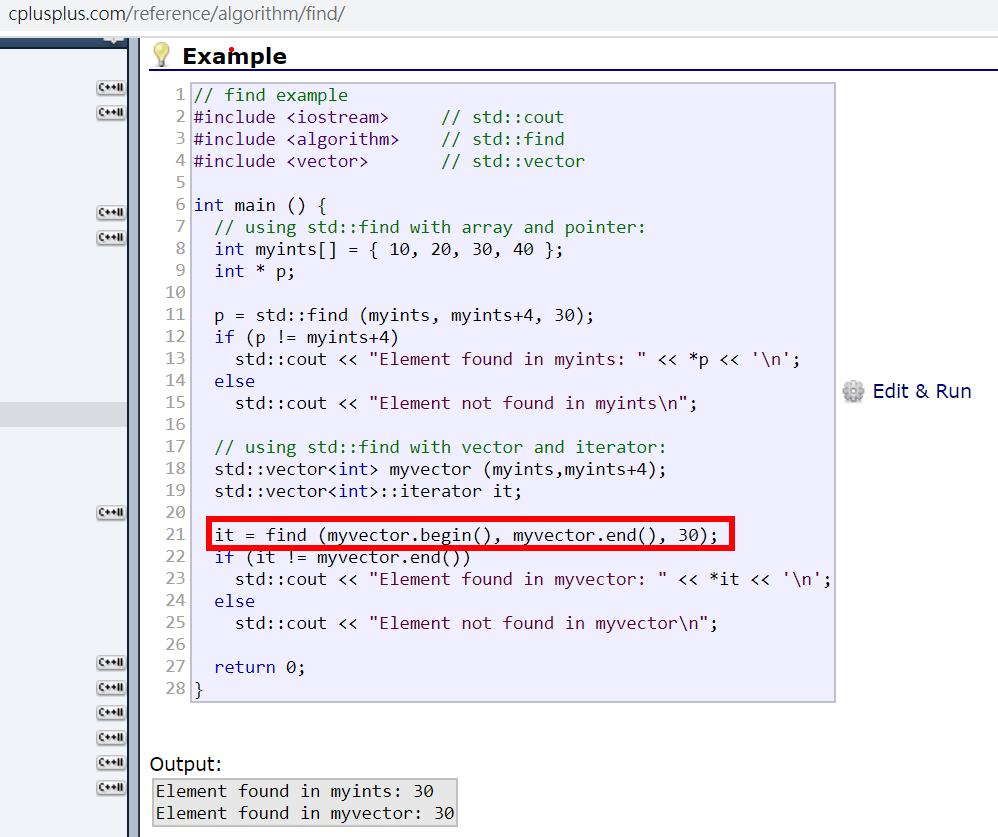






## 加分題：加入判斷是否撞牆，是否咬到自己

C++ VECTOR FIND EXAMPLE





## 加加分題：加入慣性運動(沒按鍵的時候會依照慣性前進)

