**HW3**

1. **Q&A**
2. (A) void total(int value1, int value2, int value3)

void 不會回傳，應改成int

(B) average = (value1 + value2 + value3)/3;

加上return average;

(C) void area(int length = 30 , int width)

應為(int width, int length = 30)

(D) void getValue中間要空格 (int& value)

cin >> value; 不用加&

(E) int getValue中間要空格

int inputValue; 中間要空格

return inputValue;中間要空格

double inputValue; 中間要空格

return inputValue;中間要空格

1. 關鍵詞：static 🡺 static int , retain their contents between consecutive function calls.
2. 能夠比較有條有理的清楚分析code的過程，更有邏輯且更簡約。
3. 有兩個以上的function有同樣的name
4. argument會有一個真實的值，parameter會有一個形式上的參數
5. Function prototype = before function is called we must know name, return type, number of parameters, data type of each parameter

Function head = consists of the function return type , the function name, the function parameter list

Ex.int main(parameter list)

Function definition = includes name, return type, parameter list, body.

Ex.int main(parameter list)

{ cout << “Hellow World\n”;

return 0;

}

Function call = use the function name followed by () and ;

Ex. void function();

1. 因為他會受到每一個funtion的影響，可能不小心就把它原來的值取代掉了
2. local variable他的lifetime會等同於它存在的這個funtion裡面，而global variable的lifetime包含了全部的funtion。
3. Call by value的記憶體儲存在不同的地方，call by reference和call by address 則是將變數儲存在同一個地方，只是一個分別是透過比較的方式把變數取代掉，另一個則是指定一個儲存的區塊。

EX:像是在執行變數交換的時候(swap)，就需要用到call by reference和call by address會比較方便，而call by reference 則是用起來比較方便的。

Ex. int val = 5;

Value(val)

Void value(int&) , you can work with the original data inside the calling function.

Passed by address: not output the value of function’s variables but output the value’s address in the function.

Ex. void value(int&);

void value(int& num)

{ cout << “enter a numver”;

cin >> num;

}

int localVar = 5’

value(&localVar);

1. If you have to calculate the variables or change them in your function. You have to use call by reference. If you only have to get some variables from user or display them. You can use call by value.
2. True/False

1~5 TFTTF

6~8 TFF

**C. Choice選擇題**

1~5 ACBDB

6~10 CADBD

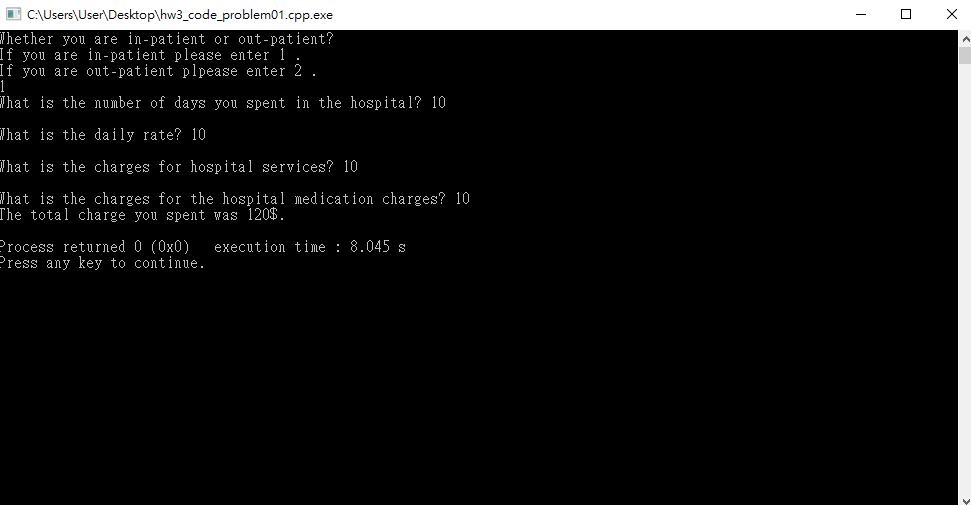
11~15 CCCEA

16~20 DBDDC

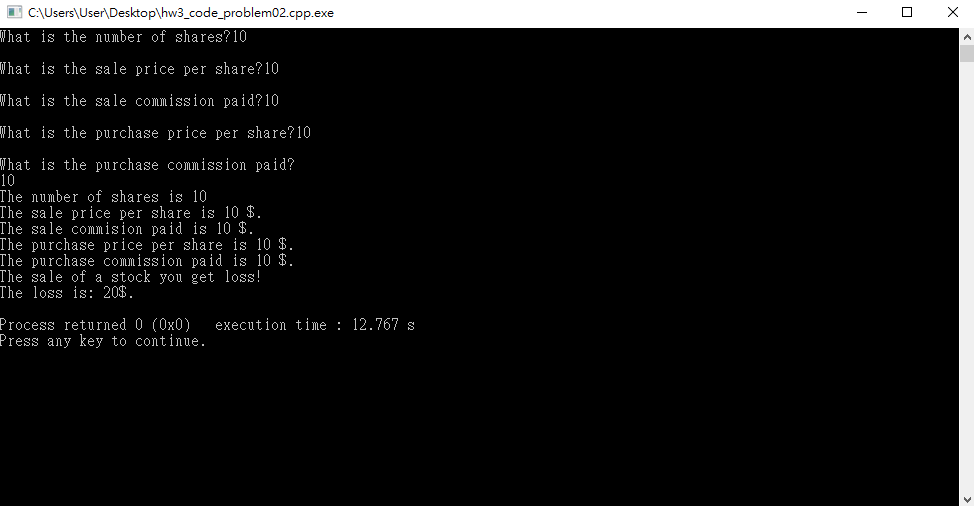
21~22 BA

**D. Programming**

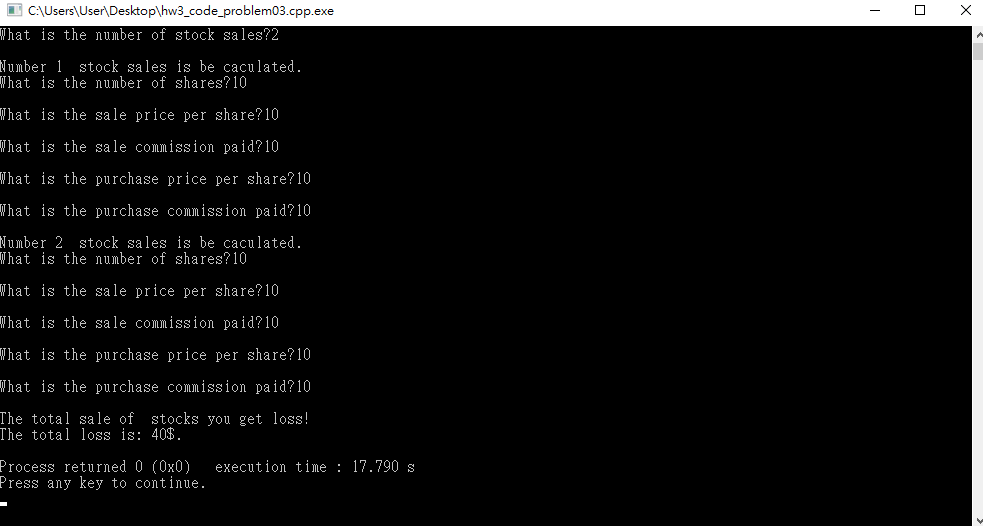
**1. Overloaded hospital**

****

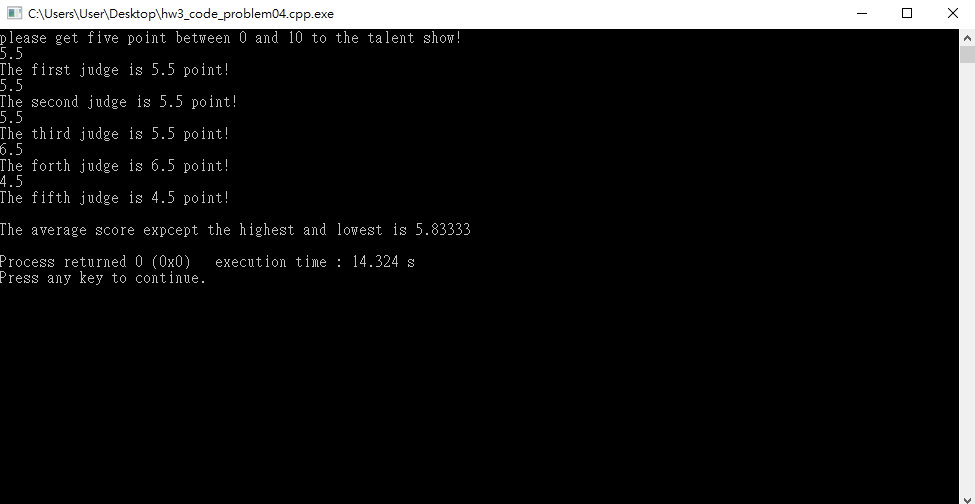
1. **Stock profit**

****

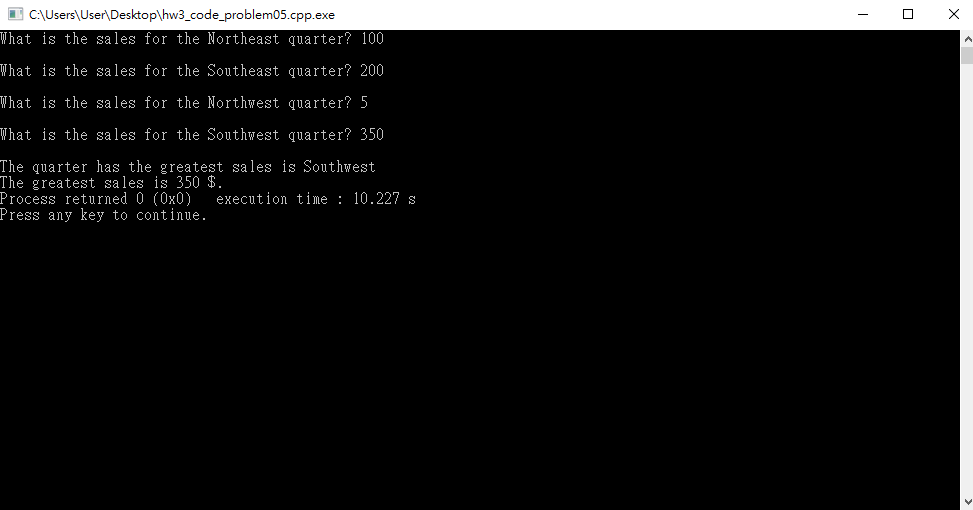
1. **Multiple stock profit**

****

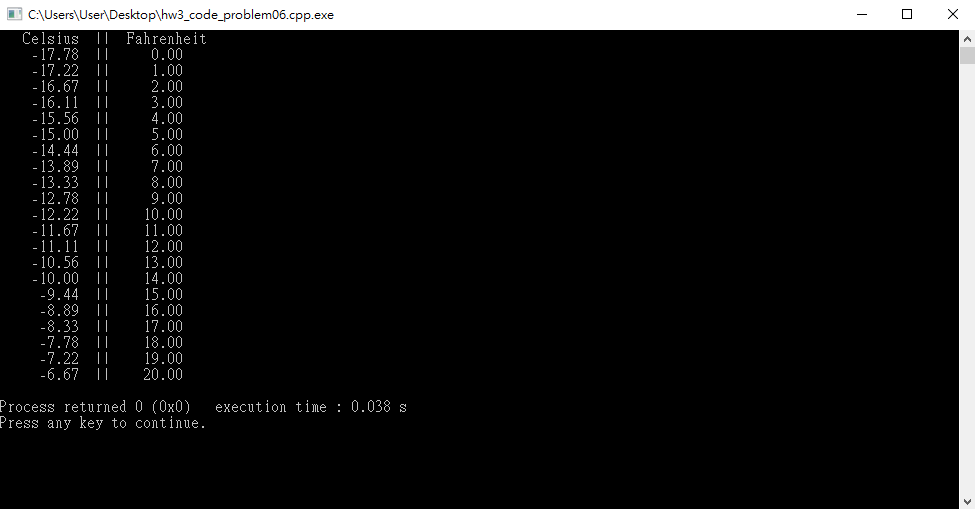
1. **Star search**

****

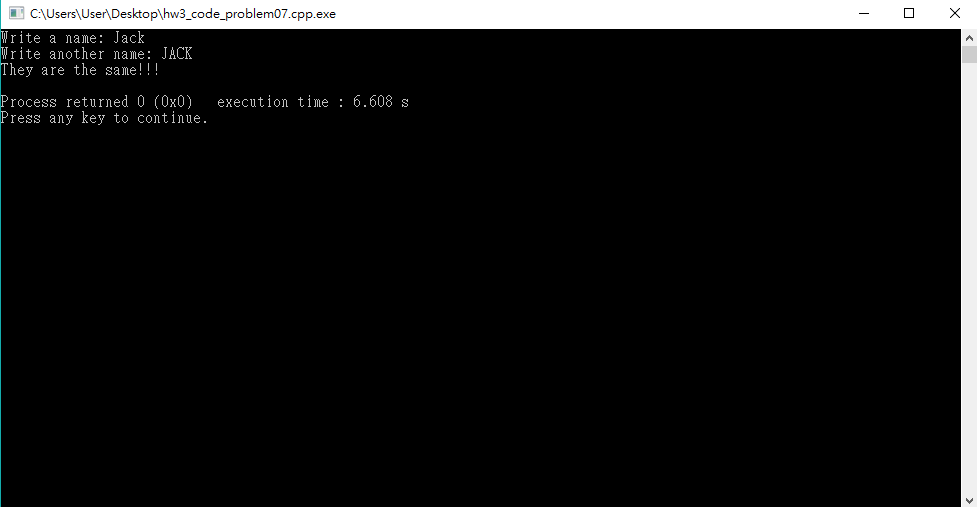
1. **Winning division**

****

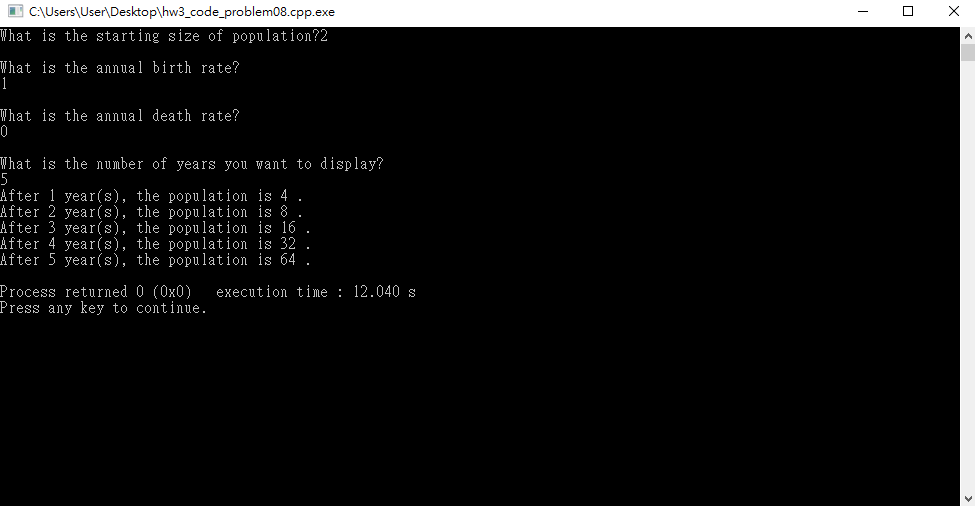
1. **Celsius Temperature Table**

****

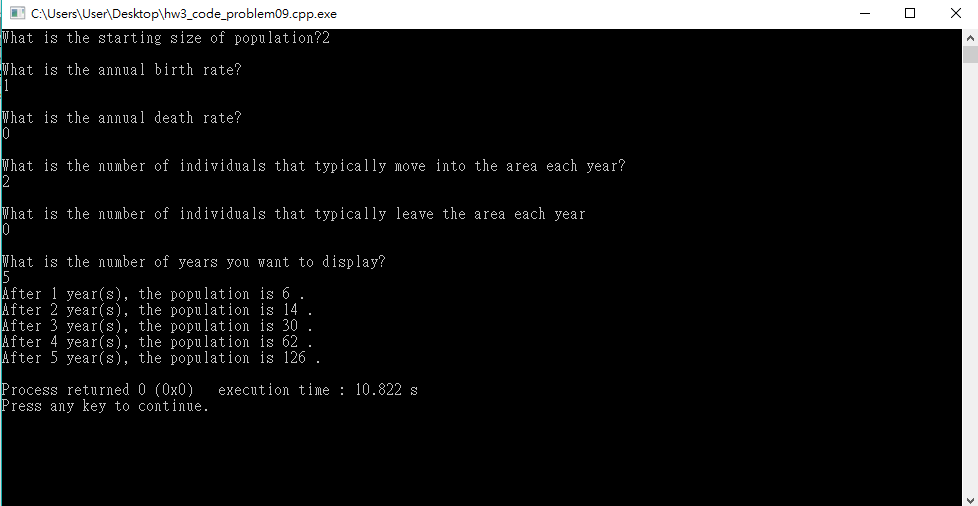
1. **String Compare**

****

1. **Population**

****

1. **Transient population**

****

**10. Input validation**

