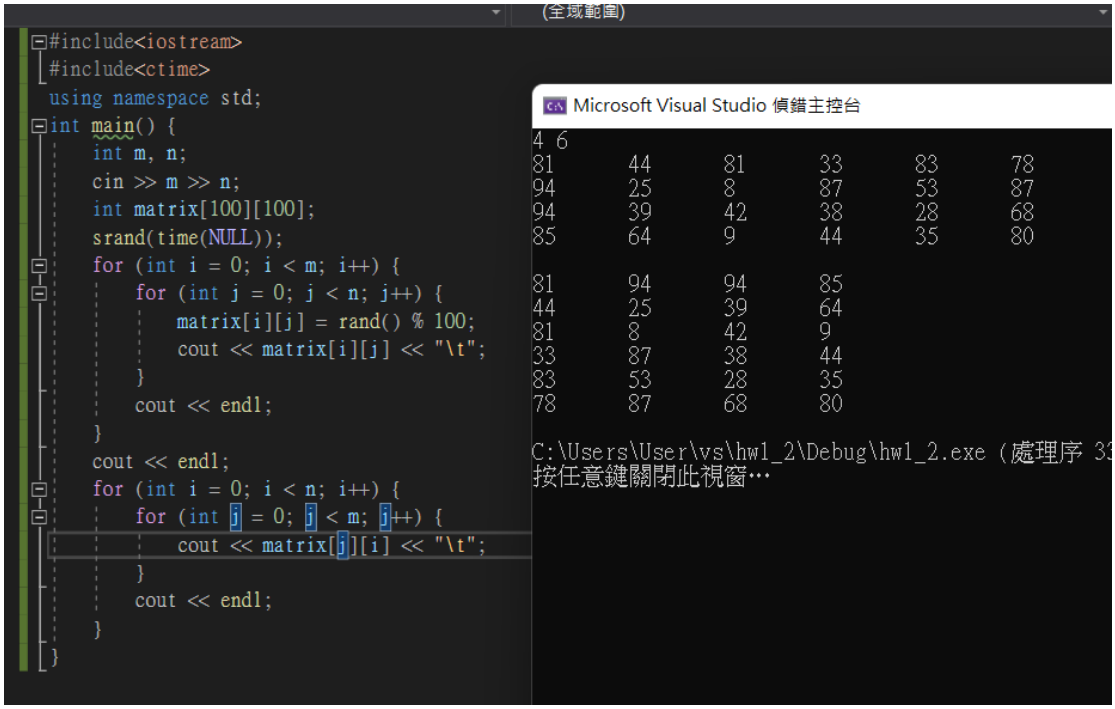
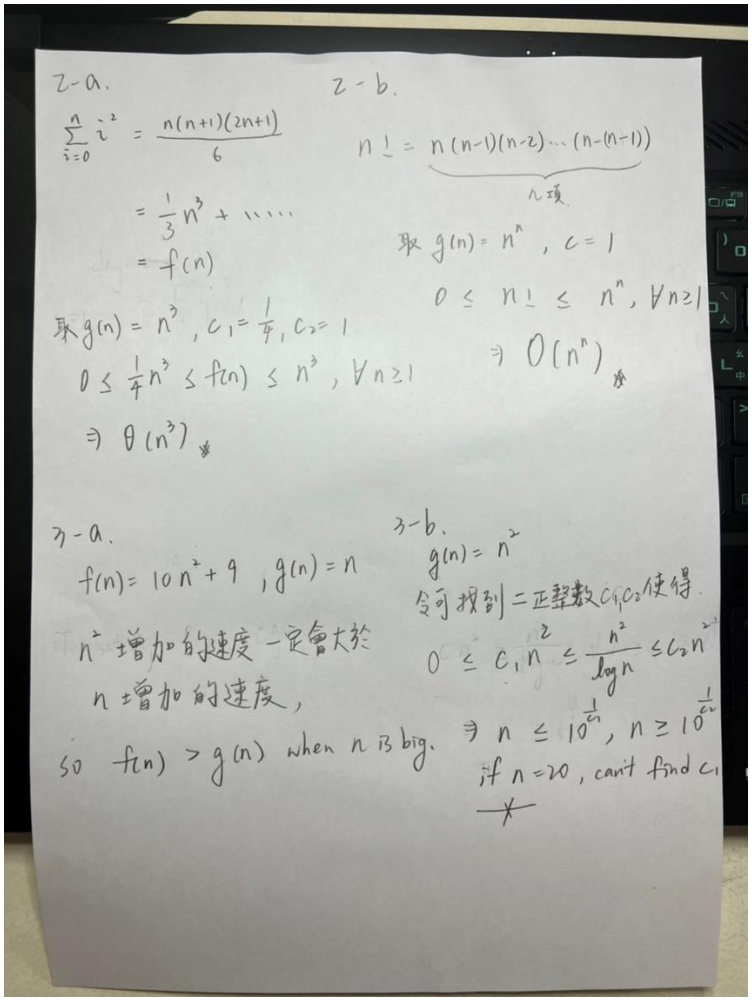
1- a


1a.cpp
hw1
(全域範圍)
main()
#include<iostream>
#include<ctime>
using namespace std;
int main() {
int n;
cin >> n;
int a[100];
srand(time(NULL));
for (int i = 0; i < n; i++) {
a[i] = rand() % 10;
cout << a[i] << "\t";
}
cout << endl << count(a, a + n,a[0]);
}
Microsoft Visual Studio 偵錯主控台
15
3 6 2 6 6 5 4 5 1 3 7 3 3 4 7
4
C:\Users\User\vs\hw1\Debug\hw1.exe (處理序 27388) 已結束，出現代碼 0。
按任意鍵關閉此視窗…
121 %

1- b


(全域範圍)
#include<iostream>
#include<ctime>
using namespace std;
int main() {
int m, n;
cin >> m >> n;
int matrix[100][100];
srand(time(NULL));
for (int i = 0; i < m; i++) {
for (int j = 0; j < n; j++) {
matrix[i][j] = rand() % 100;
cout << matrix[i][j] << "\t";
}
cout << endl;
}
cout << endl;
for (int i = 0; i < n; i++) {
for (int j = 0; j < m; j++) {
cout << matrix[j][i] << "\t";
}
cout << endl;
}
}
Microsoft Visual Studio 偵錯主控台
4 6
81 44 81 33 83 78
94 25 8 87 53 87
94 39 42 38 28 68
85 64 9 44 35 80
81 94 94 85
44 25 39 64
81 8 42 9
33 87 38 44
83 53 28 35
78 87 68 80
C:\Users\User\vs\hw1_2\Debug\hw1_2.exe (處理序
按任意鍵關閉此視窗…

2、3



4

1b.cpp 1a.cpp

w1_2

```cpp
#include<iostream>
using namespace std;
class Complex {
public:
    int r, i;
    Complex();
    Complex(int, int);
};
Complex::Complex() {
    r = 0;
    i = 0;
}

Complex::Complex(int a, int b) {
    r = a;
    i = b;
}

int main() {
    return 0;
}
```

5

```cpp
#include<iostream>
using namespace std;
class Quadratic {
public:
    int a, b, c;
    Quadratic();
    Quadratic(int,int,int);
    Quadratic operator+ (Quadratic);
};

Quadratic::Quadratic() {
    a = b = c = 0;
}
Quadratic::Quadratic(int a1, int b1, int c1) {
    a = a1;
    b = b1;
    c = c1;
}
Quadratic Quadratic::operator+(Quadratic qu) {
    Quadratic re;
    re.a = a + qu.a;
    re.b = b + qu.b;
    re.c = c + qu.c;
    return re;
}

int main() {
    Quadratic q1(1,5,9);
    Quadratic q2(2,4,-5);
    Quadratic q3;
    q3 = q1 + q2;
    cout << q3.a << " " << q3.b << " " << q3.c << " ";
}
```

6

```cpp
#include<iostream>
using namespace std;
template <class T>
class Queue {
public:
    Queue(int size = 10);
    bool IsEmpty();
    bool IsFull();
    int size();
    T& Front();
    T& Rear();
    void Push(T x);
    void Pop();
private:
    T* queue;
    int front, rear, capacity;
    int count;
};

template <class T>
Queue<T>::Queue(int size) {
    if (size < 1) throw "Queue capacity must be >0";
    queue = new T[size];
    capacity = size;
    front = 0;
    rear = -1;
    count = 0;
}

template <class T>
int Queue<T>::size() {
    return count;
}
```

```
int Queue<T>::size() {
    return count;
}

template <class T>
bool Queue<T>::IsEmpty() {
    return (size() == 0);
}

template <class T>
bool Queue<T>::IsFull() {
    return (size() == capacity);
}

template <class T>
void Queue<T>::Pop() {
    if (IsEmpty()) throw "Underflow!!!!";
    cout << "Removing " << queue[front] << endl;
    front = (front + 1) % capacity;
    count--;
}

template <class T>
void Queue<T>::Push(T item) {
    if (IsFull()) throw "Overflow!!!!";
    cout << "Inserting " << item << endl;
    rear = (rear + 1) % capacity;
    queue[rear] = item;
    count++;
}

template <class T>
inline T& Queue<T>::Front() {
```

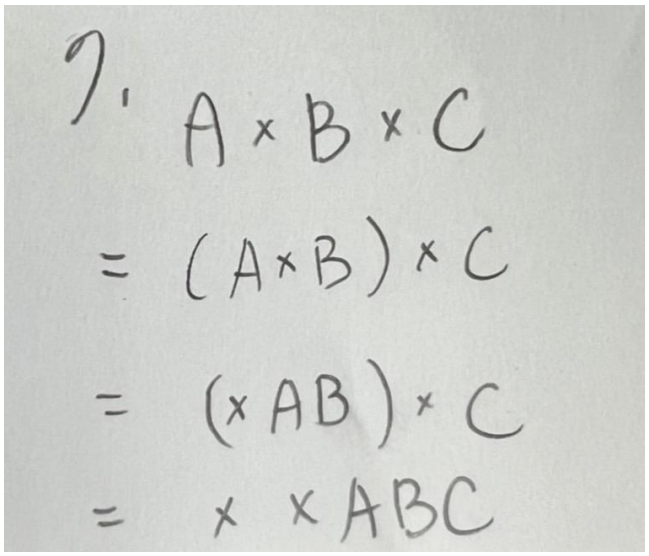
找不到任何問題

```
template <class T>
inline T& Queue<T>::Front() {
    if (IsEmpty()) throw "IsEmpty!!!";
    return queue[(front + 1) % capacity];
}

template <class T>
inline T& Queue<T>::Rear() {
    if (IsEmpty()) throw "IsEmpty!!!";
    return queue[(rear + 1) % capacity];
}

int main() {
    Queue<string> q(3);
    q.Push("a");
    q.Push("b");
    q.Push("c");

    if (q.IsFull()) cout << "q is full!\n";

    q.Pop();
    q.Pop();

    cout << q.size() << endl;

    q.Pop();
    if (q.IsEmpty()) cout << "q is empty!\n";
    //q.Pop();
}
```

Microsoft Visual Studio 偵錯主控台

```
Inserting a
Inserting b
Inserting c
q is full!
Removing a
Removing b
1
Removing c
q is empty!

C:\Users\User\vs\hw1_2\Debug\hw1_2.exe (
若要在偵錯停止時自動關閉主控台，請啟用 [
按任意鍵關閉此視窗…
```

7



8

```cpp
#include<iostream>
#include<vector>
#include<cstring>
using namespace std;
class Element {
public:
    Element(int row, int col, int val) :row(row), col(col), val(val) {};
    int row, col, val;
};

class SparseMatrix {
public:
    SparseMatrix() {}
    SparseMatrix(int row_num, int col_num ,int val_num) :row_num(row_num), col_num(col_num), val_num(val_num) {}
    SparseMatrix(const SparseMatrix& copy);
    int row_num, col_num, val_num;
    void DisplayMatrix();
    void set_Matrix();
    vector<Element> vec;
    int M[10][10];
};

SparseMatrix::SparseMatrix(const SparseMatrix& copy) {
    row_num = copy.row_num;
    col_num = copy.col_num;
    val_num = copy.val_num;
    for (int i = 0; i < 10; i++) {
```

0 5

```
25      col_num = copy.col_num;
26      val_num = copy.val_num;
27      for (int i = 0; i < 10; i++) {
28          for (int j = 0; j < 10; j++) {
29              M[i][j] = copy.M[i][j];
30          }
31      }
32      for (int i = 0; i < val_num; i++) {
33          vec.push_back(copy.vec[i]);
34      }
35  }
36
37  void SparseMatrix::DisplayMatrix() {
38      for (int i = 0; i < row_num; i++) {
39          for (int j = 0; j < col_num; j++) {
40              cout << M[i][j] << " ";
41          }
42          cout << endl;
43      }
44  }
45
46  void SparseMatrix::set_Matrix() {
47      memset(M, 0, sizeof(M));
48      for (int i = 0; i < val_num; i++) {
49          M[vec[i].row][vec[i].col] = vec[i].val;
50      }
51  }
```

```
49          M[vec[i].row][vec[i].col] = vec[i].val;
50      }
51  }
52
53  void input(SparseMatrix& SM) {
54      cin >> SM.row_num >> SM.col_num >> SM.val_num;
55      for (int i = 0; i < SM.val_num; i++) {
56          int row, col, val;
57          cin >> row >> col >> val;
58          Element element(row, col, val);
59          SM.vec.push_back(element);
60      }
61      SM.set_Matrix();
62  }
63
64  int main() {
65      SparseMatrix SM;
66      input(SM);
67      cout << "original matrix: \n";
68      SM.DisplayMatrix();
69      cout << endl << "copy matrix: \n";
70      SparseMatrix sm_copy = SM;
71      sm_copy.DisplayMatrix();
72  }
```

```
Microsoft Visual Studio 偵錯主控台
3 3 2
1 1 4
2 2 5
original matrix:
0 0 0
0 4 0
0 0 5

copy matrix:
0 0 0
0 4 0
0 0 5

C:\Users\User\vs\hw1_2\Debug\h
若要在偵錯停止時自動關閉主控台
按任意鍵關閉此視窗...
```

9

```cpp
#include<iostream>
using namespace std;
void insertionsort(int* arr, int size) {
    for (int i = 0; i < size; i++) {
        int key = arr[i];
        int j = i - 1;
        while (key < arr[j] && j>=0) {
            arr[j+1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}


void print(int* arr, int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
int main() {
    int array[7] = { 4, 2, 5, 6, 3, 9, 1 };
    print(array, 7);
    insertionsort(array, 7);
    print(array, 7);

}
```

```
Microsoft Visual Studio 偵錯主
4 2 5 6 3 9 1
1 2 3 4 5 6 9

C:\Users\User\vs\hw1_2\D
若要在偵錯停止時自動關閉
按任意鍵關閉此視窗...
```

10

```cpp
#include<iostream>
#include<string>
using namespace std;
string ispalin(string s) {
    for (int i = 0; i < s.length() / 2; i++) {
        if (s[i] != s[s.length() - i - 1]) return "NO";
    }
    return "YES";
}
int main() {
    string str;
    str = "level";
    cout << str << " " << ispalin(str) << endl;
    str = "apple";
    cout << str << " " << ispalin(str) << endl;
}
```

```
Microsoft Visual S
level YES
apple NO

C:\Users\User\vs
若要在偵錯停止時
按任意鍵關閉此視
```