9-5 實作練習

練習一：成績管理系統（基礎操作）

題目敘述

請你撰寫一個程式，功能如下：

1. 讀入 N 筆學生成績（整數，0~100）
2. 輸出所有成績
3. 輸出最高分、最低分、平均分數（取到小數點後兩位）
4. 刪除最後一筆成績後，再輸出一次所有成績

範例輸入

```
5
90 75 88 62 95
```

範例輸出

```
成績：90 75 88 62 95
最高分：95
最低分：62
平均：82.00
刪除最後一筆後：90 75 88 62
```

提示

- 使用 `push_back` 加入一筆成績
- 善用 `size()`、`pop_back()`

▼ **參考解答（請先自己嘗試！）**

```
#include <vector>
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    int n;
    cin >> n;

    vector<int> scores;
    for (int i = 0; i < n; i++) {
        int s;
        cin >> s;
        scores.push_back(s);
    }

    // 輸出所有成績
    cout << "成績：";
    for (int s : scores) cout << s << " ";
    cout << endl;

    // 找最大最小
    int maxVal = scores[0], minVal = scores[0];
    int total = 0;
    for (int s : scores) {
        if (s > maxVal) maxVal = s;
        if (s < minVal) minVal = s;
        total += s;
    }

    cout << "最高分：" << maxVal << endl;
```

```
    cout << "最低分：" << minVal << endl;
    cout << fixed << setprecision(2);
    cout << "平均：" << (double)total / scores.size() << endl;

    scores.pop_back();

    cout << "刪除最後一筆後：";
    for (int s : scores) cout << s << " ";
    cout << endl;

    return 0;
}
```

練習二：移除特定元素（insert / erase）

題目敘述

給定一個整數序列，請你：

1. 讀入 N 個整數
2. 讀入一個目標值 K
3. 移除序列中**所有等於 K 的元素**
4. 在序列**開頭**插入一個數字 0
5. 輸出最終序列

範例輸入

```
8
1 3 5 3 2 3 7 9
3
```

範例輸出

```
0 1 5 2 7 9
```

提示

- 移除元素後，迭代器的位置會改變，請注意 `erase()` 的回傳值（它會回傳被刪除元素的下一個位置）
- 在開頭插入用 `v.insert(v.begin(), 0)`

常見陷阱

```
// ❌ 錯誤寫法：erase 後 it 已失效，不可 it++
for (auto it = v.begin(); it != v.end(); it++) {
    if (*it == K) v.erase(it);
}

// ✅ 正確寫法：erase 回傳下一個有效迭代器
for (auto it = v.begin(); it != v.end(); ) {
    if (*it == K)
        it = v.erase(it);  // 不 it++
    else
        it++;
}
```

▼ 參考解答（請先自己嘗試！）

```
#include <vector>
#include <iostream>
using namespace std;

int main() {
    int n;
```

```
    cin >> n;

    vector<int> v;
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        v.push_back(x);
    }

    int K;
    cin >> K;

    // 移除所有等於 K 的元素
    for (auto it = v.begin(); it != v.end(); ) {
        if (*it == K)
            it = v.erase(it);
        else
            it++;
    }

    // 在開頭插入 0
    v.insert(v.begin(), 0);

    // 輸出
    for (int x : v) cout << x << " ";
    cout << endl;

    return 0;
}
```

練習三：矩陣加法（二維 vector）

題目敘述

給定兩個 N×M 的矩陣 A 和 B，請計算 C = A + B，並輸出結果矩陣 C。

範例輸入

```
2 3
1 2 3
4 5 6
7 8 9
1 0 1
```

（第一行為 N M，接下來 N 行為矩陣 A，再 N 行為矩陣 B）

範例輸出

```
8 10 12
5 5 7
```

提示

- 使用 `vector<vector<int>>` 來儲存矩陣
- 初始化：`vector<vector<int>> C(n, vector<int>(m, 0));`

▼ **參考解答（請先自己嘗試！）**

```
#include <vector>
#include <iostream>
using namespace std;

int main() {
    int n, m;
```

```
    cin >> n >> m;

    vector<vector<int>> A(n, vector<int>(m));
    vector<vector<int>> B(n, vector<int>(m));
    vector<vector<int>> C(n, vector<int>(m, 0));

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> A[i][j];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> B[i][j];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            C[i][j] = A[i][j] + B[i][j];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << C[i][j];
            if (j < m - 1) cout << " ";
        }
        cout << endl;
    }

    return 0;
}
```

重點總覽

常用成員函數速查表

函數	功能	時間複雜度
`push_back(val)`	在末尾新增元素	O(1) 均攤
`pop_back()`	移除末尾元素	O(1)
`v[i]`	存取索引 i 的元素（無檢查）	O(1)
`v.at(i)`	存取索引 i 的元素（有檢查）	O(1)
`size()`	回傳元素個數	O(1)
`empty()`	是否為空	O(1)
`clear()`	清空所有元素	O(n)
`front()`	第一個元素	O(1)
`back()`	最後一個元素	O(1)
`insert(it, val)`	在迭代器位置前插入	O(n)
`erase(it)`	移除迭代器位置的元素	O(n)
`resize(n)`	調整元素個數	O(n)
`reserve(n)`	預留記憶體空間	O(n)
`capacity()`	已分配的空間大小	O(1)
`begin()`	指向第一個元素的迭代器	O(1)
`end()`	指向最後一個之後的迭代器	O(1)

三大常見陷阱

1. **`erase` 後迭代器失效**：請使用 `it = v.erase(it)` 取得新的有效迭代器。
2. **`pop_back` 空 `vector`**：呼叫前先確認 `!v.empty()`。
3. **`size()` 型別為 `size_t`**：比較或相減時注意無號整數的行為。

Revision #5
Created 26 April 2026 12:41:49 by huihui
Updated 27 April 2026 07:09:41 by huihui