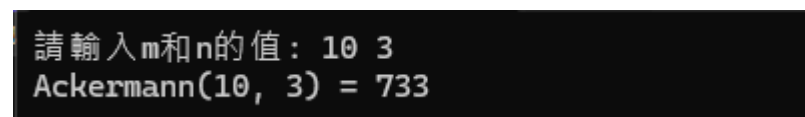


P1：

此程式再輸入過大的 M 或 N 時，答案都會出現 733 的錯誤答案(可能 733 是某個邊界情況下的錯誤計算結果)，但輸入小的數值時是沒問題的。

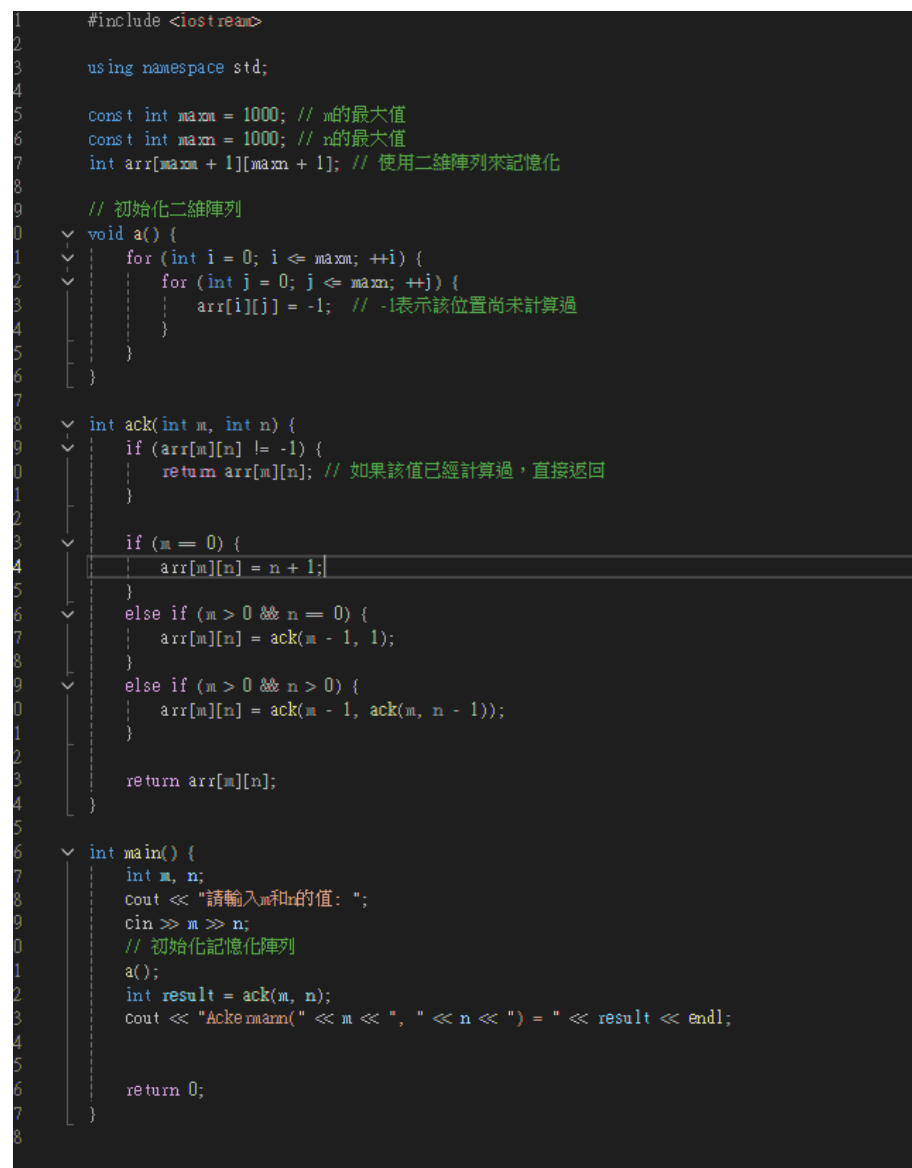


```
Microsoft Visual Studio 偵錯控制台
請輸入m和n的值： 3 2
Ackermann(3, 2) = 29
```



```
請輸入m和n的值： 10 3
Ackermann(10, 3) = 733
```

程式碼：



```
1 #include <iostream>
2
3 using namespace std;
4
5 const int maxm = 1000; // m的最大值
6 const int maxn = 1000; // n的最大值
7 int arr[maxm + 1][maxn + 1]; // 使用二維陣列來記憶化
8
9 // 初始化二維陣列
10 void a() {
11     for (int i = 0; i <= maxm; ++i) {
12         for (int j = 0; j <= maxn; ++j) {
13             arr[i][j] = -1; // -1表示該位置尚未計算過
14         }
15     }
16 }
17
18 int ack(int m, int n) {
19     if (arr[m][n] != -1) {
20         return arr[m][n]; // 如果該值已經計算過，直接返回
21     }
22
23     if (m == 0) {
24         arr[m][n] = n + 1;
25     }
26     else if (m > 0 && n == 0) {
27         arr[m][n] = ack(m - 1, 1);
28     }
29     else if (m > 0 && n > 0) {
30         arr[m][n] = ack(m - 1, ack(m, n - 1));
31     }
32
33     return arr[m][n];
34 }
35
36 int main() {
37     int m, n;
38     cout << "請輸入m和n的值： ";
39     cin >> m >> n;
40     // 初始化記憶化陣列
41     a();
42     int result = ack(m, n);
43     cout << "Ackermann(" << m << ", " << n << ") = " << result << endl;
44
45     return 0;
46 }
```

P2 :

程式碼：

```
#include <iostream>
#include <cmath>
#include <cstring>
using namespace std;
void powerset(char* set, int size) {
    unsigned int power_set_size = pow(2, size); // 計算冪集大小

    // 遍歷所有可能的子集
    for (unsigned int counter = 0; counter < power_set_size; counter++) {
        cout << "{ ";
        // 檢查每一位
        for (int j = 0; j < size; j++) {
            // 如果當前位是1，則包括這個元素
            if (counter & (1 << j)) {
                cout << set[j] << " ";
            }
        }
        cout << "}" << endl;
    }
}

int main() {
    char set[100]; // 假設最大元素數量為 100
    cout << "輸入S的內容: ";
    cin >> set;
    int size = strlen(set); // 取得集合大小
    cout << "S可能的組合為:\n";
    powerset(set, size);

    return 0;
}
```

輸入S的內容: abc

S可能的組合為：

```
{ }
{ a }
{ b }
{ a b }
{ c }
{ a c }
{ b c }
{ a b c }
```