資料結構作業

資訊工程系

姓名：康倍銓

科系：資工三甲

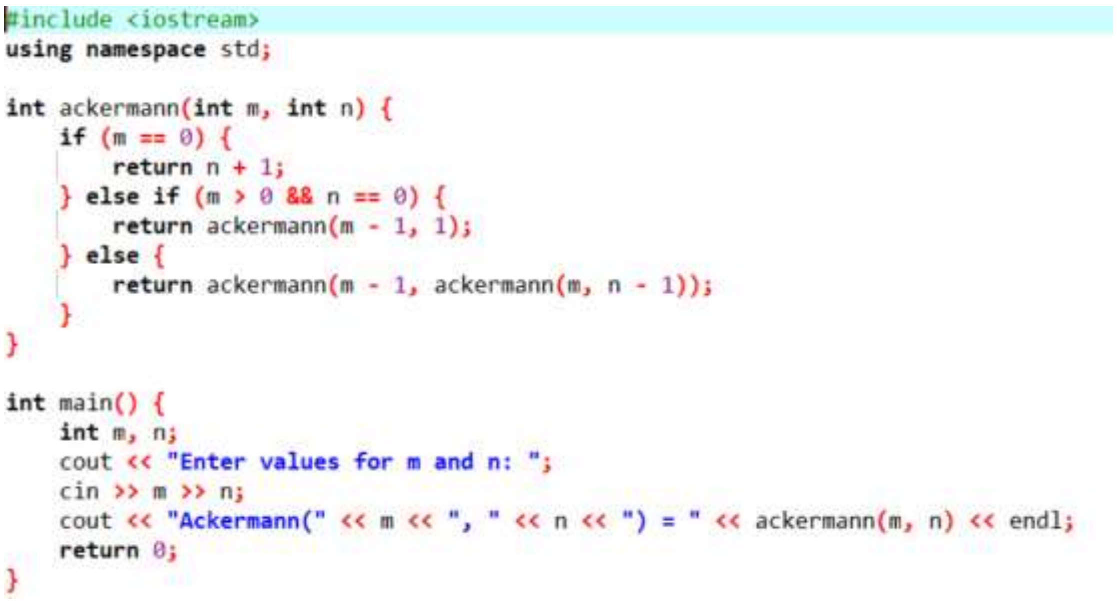
學號：41143128

指導老師：江季翰

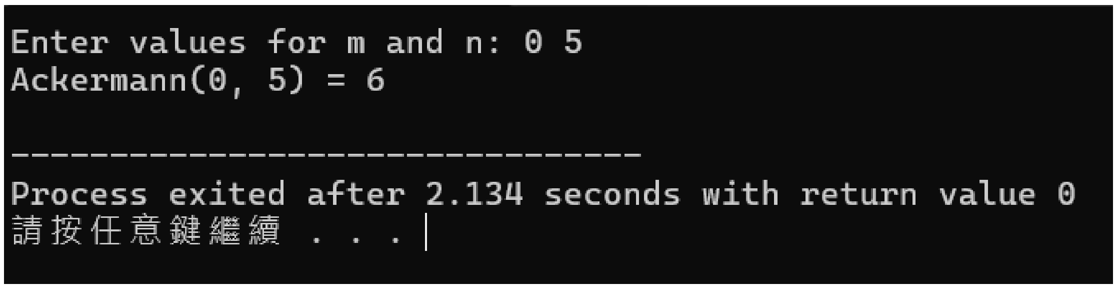
中華民國 113 年 10 月 08 日

國立虎尾科技大學
NATIONAL FORMOSA UNIVERSITY

作業一（Ackermann）

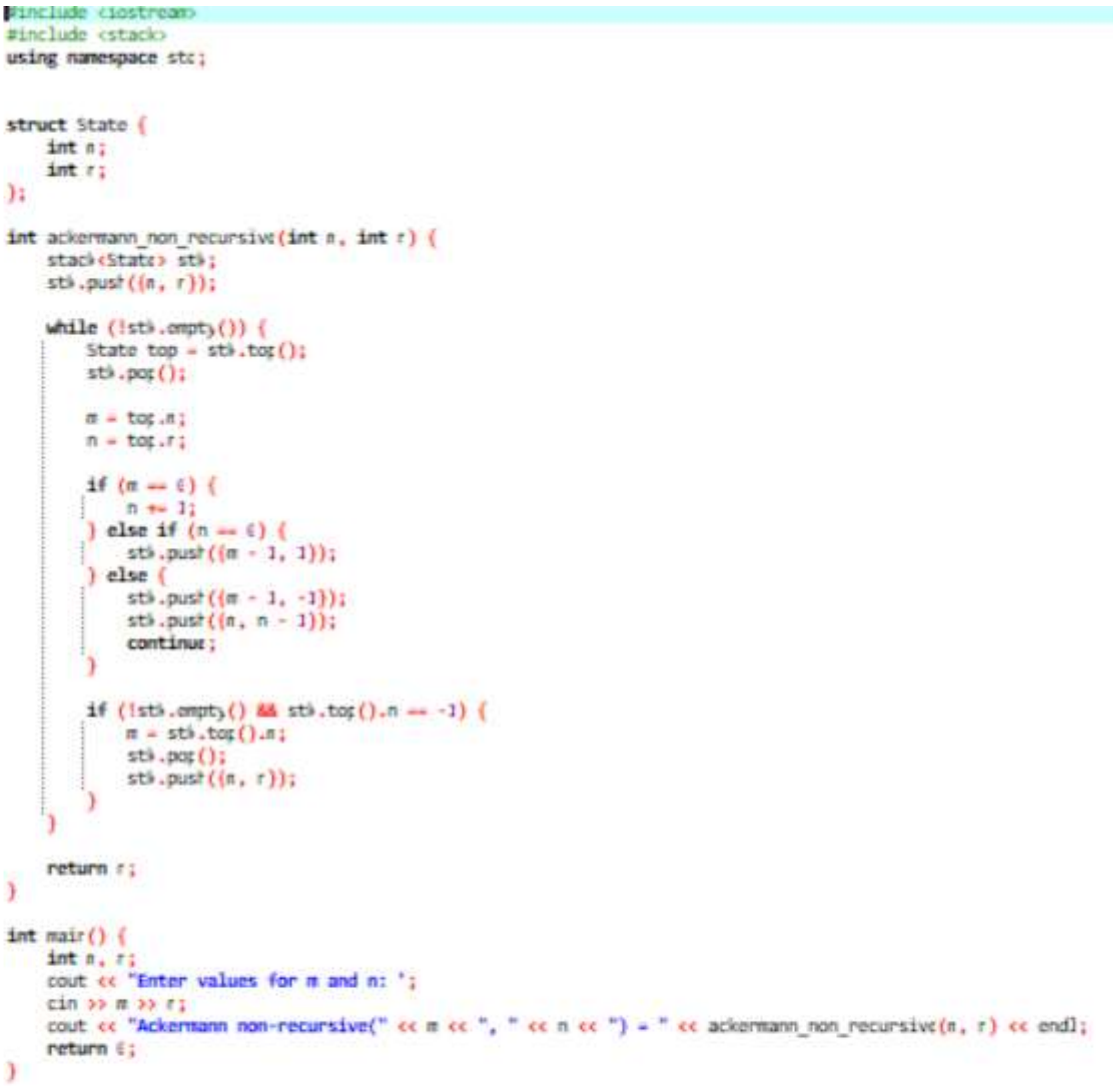


```cpp
#include <iostream>
using namespace std;

int ackermann(int m, int n) {
    if (m == 0) {
        return n + 1;
    } else if (m > 0 && n == 0) {
        return ackermann(m - 1, 1);
    } else {
        return ackermann(m - 1, ackermann(m, n - 1));
    }
}

int main() {
    int m, n;
    cout << "Enter values for m and n: ";
    cin >> m >> n;
    cout << "Ackermann(" << m << ", " << n << ") = " << ackermann(m, n) << endl;
    return 0;
}
```

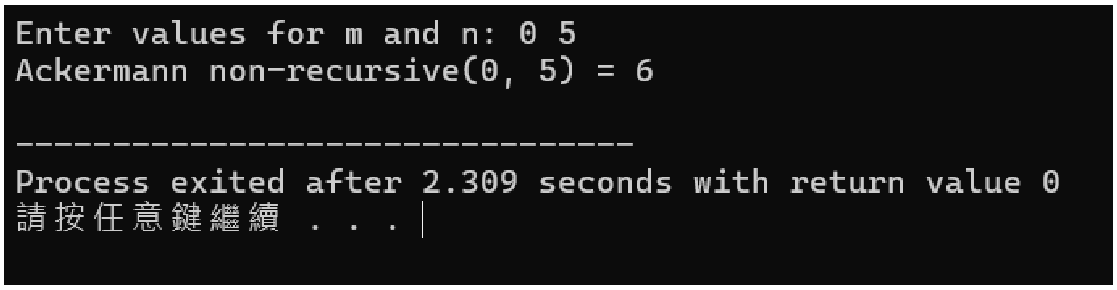


```
Enter values for m and n: 0 5
Ackermann(0, 5) = 6

--------------------------------
Process exited after 2.134 seconds with return value 0
請按任意鍵繼續 . . .
```

作業一之二（Ackermann__non）

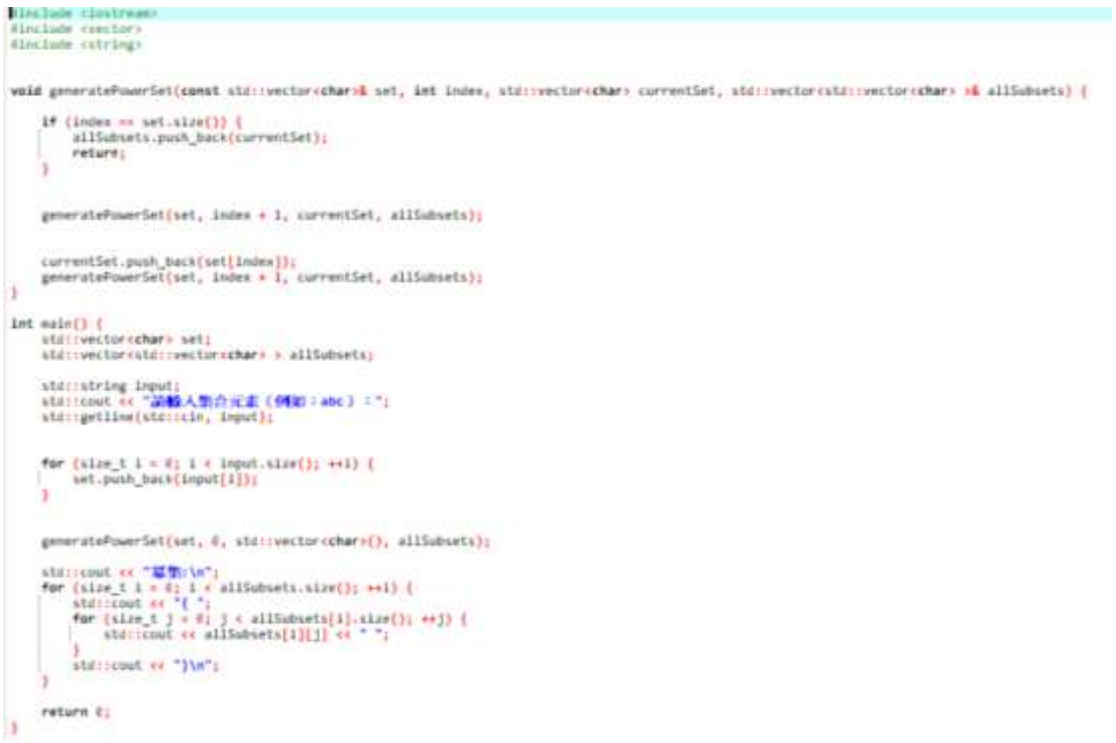


```cpp
#include <iostream>
#include <stack>
using namespace std;

struct State {
    int m;
    int n;
};

int ackermann_non_recursive(int m, int n) {
    stack<State> stk;
    stk.push({m, n});

    while (!stk.empty()) {
        State top = stk.top();
        stk.pop();

        m = top.m;
        n = top.n;

        if (m == 0) {
            n += 1;
        } else if (n == 0) {
            stk.push({m - 1, 1});
        } else {
            stk.push({m - 1, -1});
            stk.push({m, n - 1});
            continue;
        }

        if (!stk.empty() && stk.top().n == -1) {
            m = stk.top().m;
            stk.pop();
            stk.push({m, n});
        }
    }

    return n;
}

int main() {
    int m, n;
    cout << "Enter values for m and n: ";
    cin >> m >> n;
    cout << "Ackermann non-recursive(" << m << ", " << n << ") = " << ackermann_non_recursive(m, n) << endl;
    return 0;
}
```

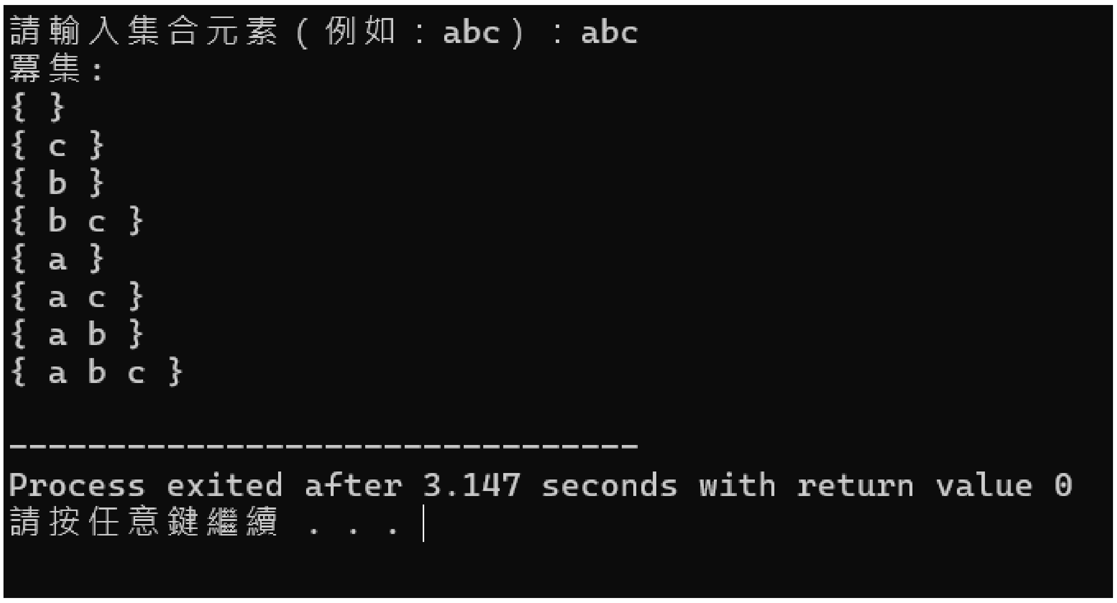


```
Enter values for m and n: 0 5
Ackermann non-recursive(0, 5) = 6

--------------------------------
Process exited after 2.309 seconds with return value 0
請按任意鍵繼續 . . .
```

作業二（降冪）



```cpp
#include <iostream>
#include <vector>
#include <string>

void generatePowerSet(const std::vector<char>& set, int index, std::vector<char> currentSet, std::vector<std::vector<char> >& allSubsets) {

    if (index == set.size()) {
        allSubsets.push_back(currentSet);
        return;
    }


    generatePowerSet(set, index + 1, currentSet, allSubsets);


    currentSet.push_back(set[index]);
    generatePowerSet(set, index + 1, currentSet, allSubsets);
}

int main() {
    std::vector<char> set;
    std::vector<std::vector<char> > allSubsets;

    std::string input;
    std::cout << "請輸入集合元素（例如：abc）：";
    std::getline(std::cin, input);


    for (size_t i = 0; i < input.size(); ++i) {
        set.push_back(input[i]);
    }


    generatePowerSet(set, 0, std::vector<char>(), allSubsets);

    std::cout << "冪集:\n";
    for (size_t i = 0; i < allSubsets.size(); ++i) {
        std::cout << "{ ";
        for (size_t j = 0; j < allSubsets[i].size(); ++j) {
            std::cout << allSubsets[i][j] << " ";
        }
        std::cout << "}\n";
    }

    return 0;
}
```



```
請輸入集合元素（例如：abc）：abc
冪集:
{ }
{ c }
{ b }
{ b c }
{ a }
{ a c }
{ a b }
{ a b c }

--------------------------------
Process exited after 3.147 seconds with return value 0
請按任意鍵繼續 . . .
```