HW4

Depth Determination problem

B10532027 四電資二 林科廷

程式摘要 ：

由 C++撰寫, 依照輸入指令, 對 node 做操作

程式本體 & 註解 ：

```
#include <iostream>
#include <vector>

using namespace std;

class node {
public:
     void SetParent(node* p) {//設定母節點
          depth = p->depth + 1;

          for (int i = 0; i < child.size(); i++) {
               child[i]->SetParent(this);
          }
     }
;

//private:

     int number;
     int depth;
     vector<node*>child;
     node* parent;

};

int main() {
```

```
node roots[999];
int rootsize = 0;
char instruction = 'x';
int number1 = -1;
int number2 = -1;

while (cin >> instruction) {


	if (instruction == 'M') {//新增node
		cin >> number1;
		roots[number1].number = number1;
		roots[number1].depth = 0;
		if (rootsize < number1)rootsize = number1;
	}
	else if (instruction == 'F') {//印出node的深度
		cin >> number1;
		cout << number1 << " " << roots[number1].depth << endl;

	}
	else if (instruction == 'G') {//合併node
		cin >> number1;
		cin >> number2;

		roots[number2].child.push_back(&roots[number1]);
		roots[number1].parent = &roots[number2];
		roots[number1].SetParent(&roots[number2]);
	}
	else if (instruction == 'E') {//結束
		break;
	}
	//cout << rootsize << endl;
	//for (int i = 0; i <  rootsize; i++) {
	//	cout <<"number : "<< roots[i].number << endl;
	//	cout <<"depth : "<< roots[i].depth << endl;
	//	cout << "child : ";
	//	for (int j = 0; j < roots[i].child.size(); j++) {
	//		cout << roots[i].child[j]->number << " ";
```

```
        //  }
        //  cout << endl;
        //  //cout <<"parent : "<< roots[i].parent->number << endl;
        //  cout << endl;
        //  cout << endl;
        //}
    }

    //system("pause");
    return 0;
    }
```

Pseudo code:

If(input==m)

Node[number].depth=0;

If(input==f)

Printf(Node[number].depth)

If(input==g)

Node[number].setparent(Node[number2]);

If(input==e)

Break;

圖解　：