

# Phill-DS-1025

Stack 例題 (用之前的 class)

括號 match

- ((( xxx ))) basic
- ((a+b)\*(c+d)) → yes
- (a+b)\*(c+d)) → no

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

struct Node{
    int data;
    Node* next;
};

class StackList{
private:
    Node* top;

public:
    StackList() : top(nullptr) {}

    void push(int data){
        Node* newNode = new Node;
        newNode->data = data;
        newNode->next = top; //新增的節點在左邊
        top = newNode; //指向最後加入的新節點
    }

    int pop(){
        if(top == nullptr){//沒東西 pop
            cout << "stack underflow" << endl;
            return -9999;
        }
        Node* current = top; //暫存要刪掉的節點
        top = top-> next; //把top 位置先調到下一個
        int value = current-> data; //先存return 值
        delete current;
        return value;
    }
}
```

```

bool isEmpty(){
    return (top==nullptr);
}

~StackList(){
    while(!isEmpty()){
        pop(); //會照 traverse 方式 free
    }
}

};

int main()
{
    string expression="(a+b)*(c+d)";
    StackList st;
    bool flag = true;

    for(char i : expression) { //遍歷一次從左到右
        if(i=='('){
            st.push(i); //將來要找到 ) 才滿足條件
        }
        else if(i==')'){
            if(st.isEmpty()){
                flag=false;
                break; //一個不正確 後面不用比了
            }
            st.pop(); //消掉前面一個push 的 (
        }
        //其他符號不重要
    }

    if(flag==true)
        cout << "yes!"<< endl;
    else
        cout << "no!"<< endl;

    return 0;
}

```

## 後序表示式

8+7 → infix notation

87+ → postfix notation

- 用個位數字來做運算

- ex:  $23+82-* \rightarrow = -30$
- 輸入一個postfix expression  $\rightarrow$  答案
- 條件: 假設輸入的express 合乎格式(不須檢查格式)
- 條件: 只處理 + - \* /

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

struct Node{
    int data;
    Node* next;
};

class StackList{
private:
    Node* top;

public:
    StackList() : top(nullptr) {}

    void push(int data){
        Node* newNode = new Node;
        newNode->data = data;
        newNode->next = top; //新增的節點在左邊
        top = newNode; //指向最後加入的新節點
    }

    int pop(){
        if(top == nullptr){//沒東西 pop
            cout << "stack underflow" << endl;
            return -9999;
        }
        Node* current = top; //暫存要刪掉的節點
        top = top-> next; //把top 位置先調到下一個
        int value = current-> data; //先存return 值
        delete current;
        return value;
    }

    bool isEmpty(){
        return (top==nullptr);
    }

    ~StackList(){
        while(!isEmpty()){
            pop(); //會照 traverse 方式 free
        }
    }
};
```

```

    }
}
};

int main()
{
    string expression="23+82-*";
    StackList st;

    for(char i : expression){
        if(isdigit(i)){//是數字
            st.push(i-'0');
        }
        else{ //不是數字, 合於規格, 一定運算子
            int op1=0, op2=0;
            switch(i){
                case '+':
                    op1 = st.pop();
                    op2 = st.pop();
                    st.push(op1+op2); // 因為結果會是下一層的 operand
                    break;
                case '-':
                    op1 = st.pop();
                    op2 = st.pop();
                    st.push(op1-op2); // 因為結果會是下一層的 operand
                    break;
                case '*':
                    op1 = st.pop();
                    op2 = st.pop();
                    st.push(op1*op2); // 因為結果會是下一層的 operand
                    break;
                case '/':
                    op1 = st.pop();
                    op2 = st.pop();
                    st.push(op1/op2); // 因為結果會是下一層的 operand
                    break;
            }
        }
    }

    cout << "final result=" << st.pop() << endl;
    return 0;
}

```