1121DS TA #02

TAs：陳柏翰、藍振恩、朱孟淇、李冠穎

Outline

- 規則
- PracticeA
- PracticeD
- 中序轉後序

規則

通過 ZeroJudge 系統 (網頁型judge評分工具) 的評測，使用瀏覽器即可submit程式碼。登入後右上角總覽點選參加課程並輸入代碼。

- ZeroJudge網址：https://zerojudge.tw/
- 課程網址：https://zerojudge.tw/ShowVClass?vclassid=2073
- 課程代碼：**rGJIKs**
- 請在 ZeroJudge 系統裡設定自己的**公開暱稱**為登入moodle的帳號 (舉例: Z91234567，z_201234)，未設定成規定格式導致成績無法登陸時**後果自負**。
- 其他規則詳見 Moodle 文件。

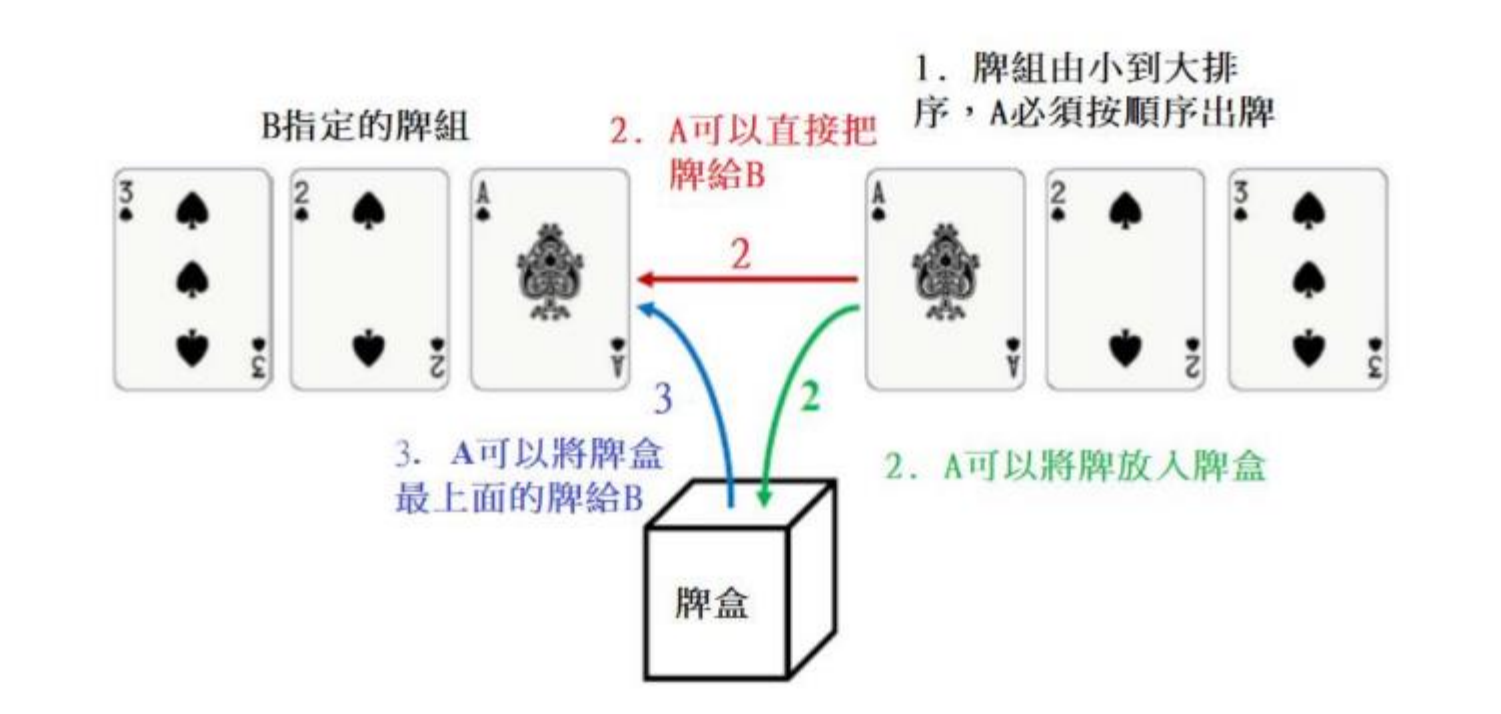
PracticeA

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

int main() {
    int time;
    cin>>time; // 測試 M 個數字
    while (time > 0){
        string s;
        cin>>s;
        int len = s.length();
        int odd = 0;
        int even = 0;
        // 比較奇位數與偶為數總和是否一樣
        for (int i = 0; i < len; i++){
            if (i%2 == 0) even = even + s[i] - '0';
            else odd = odd + s[i] - '0';
        }
        // 輸出
        if (even == odd) cout<<"YES\n";
        else cout<<"NO\n";
        time = time - 1;
    }
    return 0;
}
```

PracticeD

- A和B今天在玩一種撲克牌遊戲，一開始A有一副排序為1~3且數字不重複的牌(1,2,3)，A勝利的條件是需按順序交出B指定的牌。



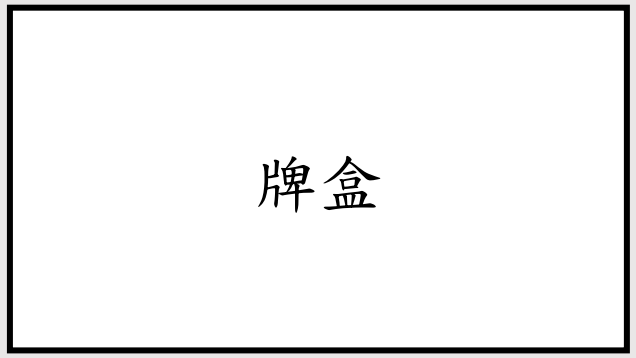
PracticeD Example1

Need:1 2 3

B

A

1 2 3



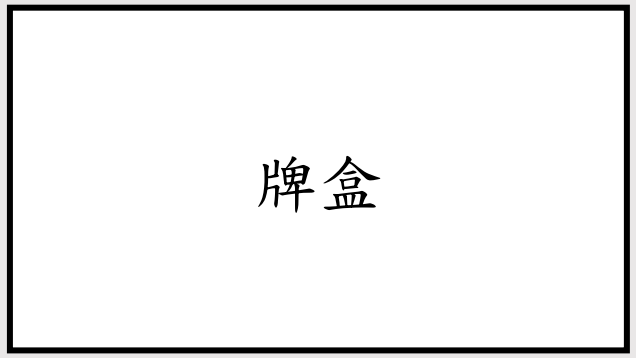
PracticeD Example1

Need:1 2 3

B

A

2 3



1

PracticeD Example1

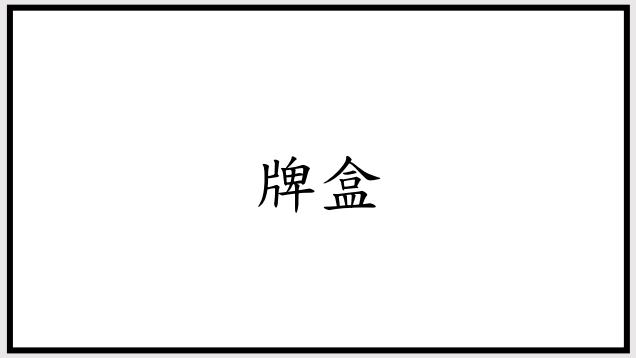
Need:1 2 3

B

1

A

2 3



PracticeD Example1

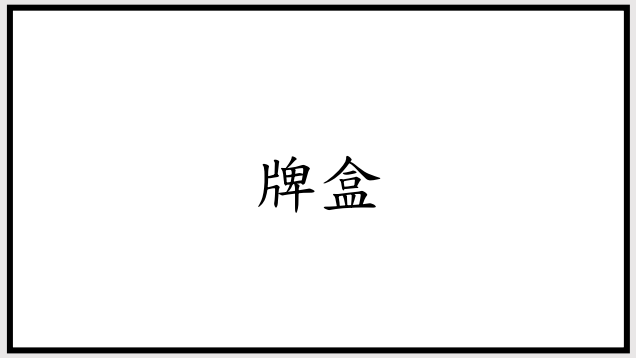
Need:1 2 3

B

A

1

3



2

PracticeD Example1

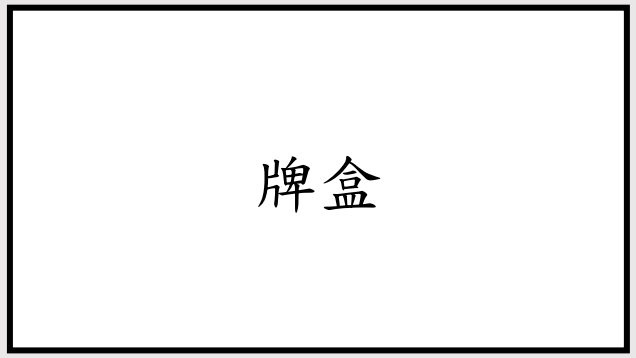
Need:1 2 3

B

1 2

A

3

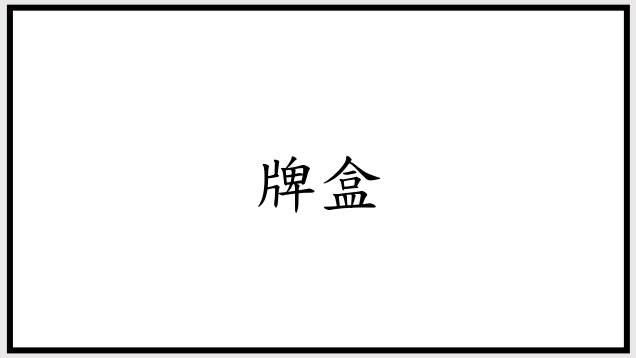


PracticeD Example1

Need:1 2 3

B A

1 2



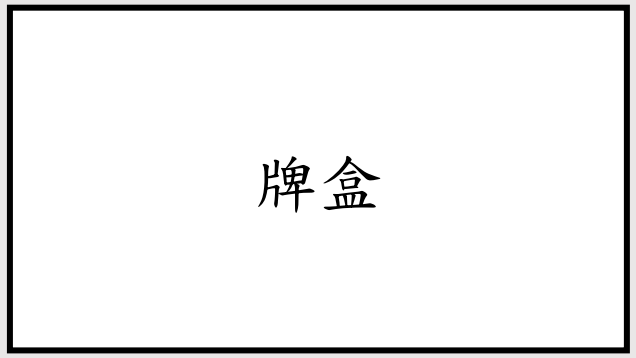
3

PracticeD Example1

Need:1 2 3

B Yes A

1 2 3



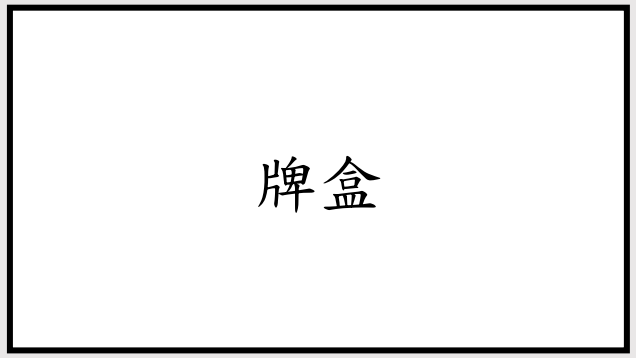
PracticeD Example2

Need:3 1 2

B

A

1 2 3



PracticeD Example2

Need:3 1 2

B A

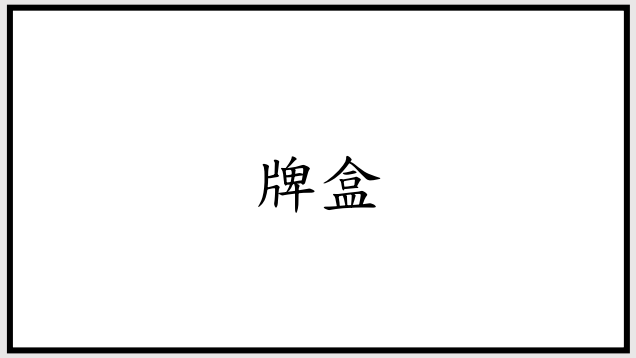
3
2
1

PracticeD Example2

Need:3 1 2

B A

3



2

1

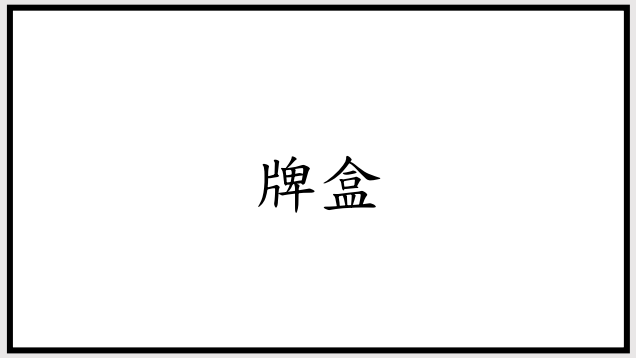
PracticeD Example2

Need:3 1 2

B A

No

3



2

1

Top不是1

PracticeD

```cpp
#include<iostream>
#include<stack>
using namespace std;
int main()
{
    stack<int> st;
    int idx=1,b[100];
    for(int i=1;i<=3;i++)
        cin>>b[i];
    for(int i=1;i<=3;i++)
    {
        st.push(i);
        while(!st.empty() && st.top()==b[idx])
            idx++ , st.pop();
    }
    if(idx==4)
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;
}
```

中序轉後序

- 演算法意義匯整:
  1. **中序運算式由左往右掃描，當遇到:**
     - **1-1. 運算元**: 直接輸出 (或Print) 到後序式
     - **1-2. 運算子**:
       - **1-2-1. ")"**: pop堆疊內的運算子直到遇到 "("
       - **1-2-2. 其它運算子x**: 比大小
         1. 若運算子x 的優先權 > 堆疊內最Top的運算子時，則將運算子x push至堆疊中
         2. 若運算子x的優先權 ≤ 堆疊內最Top的運算子時，則pop堆疊內的運算子直到x > 堆疊內最Top的運算子為止
  2. **掃描完中序運算式，則將堆疊內的殘餘資料pop完**
- Note:
  - Stack為空時，其優先權最低。(∵Stack沒有任何運算子可與待輸入的運算子做比較!!)
  - "(" 在Stack外優先權最高，但在Stack內優先權最低。

中序轉後序 (a + b) * (c + d)

元素	堆疊	輸出
(	(	-
a	(	a
+	(+	a
b	(+	ab
)	-	ab+
*	*	ab+
(	*(	ab+
c	*(	ab+c
+	*(+	ab+c
d	*(+	ab+cd
)	*	ab+cd+
-	-	ab+cd+*

中序轉後序

```
int getPrecedence(char op) {
    if (op == '+' || op == '-') return 1;
    if (op == '*' || op == '/') return 2;
    return 0;
}
```

```
int main()
{
    string infix_expression;
    cout << "請輸入中序表達式：";
    cin >> infix_expression;

    string postfix_expression = infixToPostfix(infix_expression);
    cout << "後序表達式為：" << postfix_expression << endl;

    return 0;
}
```

```
string infixToPostfix(const string& infix)
{
    string postfix = "";
    stack<char> s;

    for (int i = 0; i < infix.length(); i++)
    {
        char c = infix[i];
        if (isalnum(c)) //遇到字母直接輸出
        {
            postfix += c;
        }
        else if (c == '(') //遇到左括號加進堆疊
        {
            s.push(c);
        }
        else if (c == ')') //遇到右括號
        {
            while (!s.empty() && s.top() != '(') //若推疊不是空且頂端不是左括號
            {
                postfix += s.top(); //把頂端的輸出
                s.pop(); //pop the top
            }
            s.pop(); // pop the '('
        }
        else //遇到運算子
        {
            // c的運算優先級小於等於top時，把top輸出再把c加入堆疊
            while (!s.empty() && getPrecedence(c) <= getPrecedence(s.top()))
            {
                postfix += s.top();
                s.pop();
            }
            s.push(c);
        }
    }

    while (!s.empty())
    {
        postfix += s.top();
        s.pop();
    }

    return postfix;
}
```

Thanks