

ZeroJudge : 基礎練習

1. a001. 哈囉

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

main(){
    string s = "";
    cin >> s;
    cout << "hello," << endl << s << endl;
}
```

2. a003. 兩光法師占卜術

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

int main() {
    int month, day;
    cin >> month >> day;

    int S = (month * 2 + day) % 3;

    if (S == 0) cout << "普通";
```

```
else if (S == 1) cout << "吉";
else cout << "大吉";

return 0;
}
```

3. a004. 文文的求婚

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

int main() {
    int year;
    while (cin >> year) {
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
            cout << "閏年" << endl;
        else
            cout << "平年" << endl;
    }
    return 0;
}
```

4. a005. Eva 的回家作業

說明：給一定個陣列，已知前四項，推估出第 5 向(等比或等差)

舉例來說： $20=2 \times 2 \times 5 \Rightarrow$ 可以拆成【因數】跟【次方】

```

#include<iostream>
using namespace std;
int main(){
    int t;
    cin>>t;
    for(int i=0;i<t;i++){
        int A[4];
        for(int j=0;j<4;j++){
            cin>>A[j];
            if(A[3]-A[2]==A[2]-A[1])
                cout<<A[0]<<endl<<A[1]<<endl<<A[2]<<endl<<A[3]<<endl<<A[3]+(A[3]-A[2])<<endl;
            else if(A[3]/A[2]==A[2]/A[1])
                cout<<A[0]<<endl<<A[1]<<endl<<A[2]<<endl<<A[3]<<endl<<int(A[3]*A[3]/A[2])<<endl;
        }
    }
    return 0;
}

```

5. a022. 迴文

```

def is_palindrome(s):
    return s == s[::-1]

# 讀入輸入字串
text = input().strip()

```

判斷並輸出結果

```
print("yes" if is_palindrome(text) else "no")
```

6. a024. 最大公因數(GCD)

找出最大公因數 => 輾轉相除法

$GCD(12, 15)$

→ $GCD(15, 12)$ // 先交換

→ $GCD(12, 3)$ // $15 \% 12 = 3$

→ $GCD(3, 0)$ // $12 \% 3 = 0$

→ 結果是 3

→ $GCD(a, b) = GCD(b, a \% b)$

6.1. 解法 01：用迴圈的角度

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

int main() {
    int a, b;
    while (cin >> a >> b) {
        while(a%b!=0){
            int k = a%b;
            a = b;
            b = k;
        }
    }
}
```

```
        cout << b << endl;
    }
}
```

解法 02：用遞迴的角度

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

// 計算最大公因數（遞迴版）
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}

int main() {
    int a, b;
    while (cin >> a >> b) {
        cout << gcd(a, b) << endl;
    }
    return 0;
}
```

7. a034. 二進位制轉換

將 10 進為轉成 2 進位。我們先模擬一遍

$$6 \div 2 = 3 \dots 0$$

$$3 \div 2 = 1 \dots 1$$

$$1 \div 2 = 0 \dots 1$$

餘數反過來 => 110

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

int main() {
    int num;
    while (cin >> num) {
        string ans = "";
        while (num > 0) {
            // ans += num % 2 + '0';
            char bit = char(num % 2 + '0');
            ans = bit + ans;

            num /= 2;
        }
        cout << ans << endl;
    }
}
```

8. a058. MOD3

一行輸出三個整數（空格分開），依序是：

- `mod 3 == 0` 的個數
- `mod 3 == 1` 的個數
- `mod 3 == 2` 的個數

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

int main() {
    int n;
    cin >> n;

    int count0 = 0; // 3 的倍數
    int count1 = 0; // 餘 1
    int count2 = 0; // 餘 2

    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;

        if (x % 3 == 0)
            count0++;
        else if (x % 3 == 1)
            count1++;
        else
            count2++;
    }
}
```

```
cout << count0 << " " << count1 << " " << count2 << endl;
return 0;
}
```

9. a059. 完全平方和

給你一個範圍 a 到 b ，請你找出 a 與 b 之間所有完全平方數的和。

e.g. 範圍 3-25，表示 3 至 25 中所有完全平方數的和就是 $4 + 9 + 16 + 25 = 54$ 。

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

int main() {
    int T;
    cin >> T;

    for (int caseNum = 1; caseNum <= T; ++caseNum) {
        int a, b, sum = 0;
        cin >> a >> b;

        ///// 從 sqrt(a) 到 sqrt(b) 中的整數，檢查平方數 /////
        int start = ceil(sqrt(a));
        int end = floor(sqrt(b));
        for (int i = start; i <= end; ++i) {
            sum += i * i;
        }
    }
}
```



```
    cout << "Case " << caseNum << ": " << sum << endl;
}
return 0;
}
```

10. a121. 質數又來囉

針對每組輸入的區間 $[a, b]$ （保證 $b-a \leq 1000$ ）計算範圍內的質數個數。

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

// 判斷一個數是否為質數
bool isPrime(int n) {
    if (n < 2) return false;
    if (n == 2) return true;
    if (n % 2 == 0) return false;
    for (int i = 3; i < sqrt(n)+1; i += 2) {
        if (n % i == 0) return false;
    }
    return true;
}

int main() {
    int a, b;
    while (cin >> a >> b) {
```

```
int count = 0;
for (int i = a; i <= b; ++i) {
    if (isPrime(i)) count++;
}
cout << count << endl;
}
return 0;
}
```

11. a147. Print it all

會有多組整數 n ，每組一行

每次 n 表示要輸出「所有小於 n 且不能被 7 整除的正整數」

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

int main() {
    int n;
    while (cin >> n && n != 0) {
        for (int i = 1; i < n; i++) {
            if (i % 7 != 0) {
                cout << i;
                if (i != n - 1) cout << " ";
            }
        }
        cout << endl;
    }
}
```

```
}  
return 0;  
}
```

12. a148. You Cannot Pass?!

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int n;  
    while (cin >> n && n != 0) {  
        int sum = 0, score;  
        for (int i = 0; i < n; i++) {  
            cin >> score;  
            sum += score;  
        }  
        double avg = sum * 1.0 / n; // 避免整數除法(重要)  
        if (avg > 59)  
            cout << "no" << endl;  
        else  
            cout << "yes" << endl;  
    }  
    return 0;  
}
```

13. a215. 明明愛數數

給兩個數字， n 跟 m 。試問 n 、 $n+1$ 、 $n+2$ 、...，相加到多少會超過 m 。

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

int main() {
    int n, m;
    while (cin >> n >> m) {
        int sum = 0;
        int count = 0;
        while(true){
            count++;
            sum += n;
            n++;
            if(sum>m) break;
        }
        cout << count << endl;
    }
    return 0;
}
```

14. a248. 新手訓練～陣列應用

```
#include <iostream>
#include <string>
```

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

void divide(int a, int b, int N) {
    string result;
    int integerPart = a / b;
    result = to_string(integerPart) + ".";

    a = a % b;
    for (int i = 0; i < N; ++i) {
        a *= 10;
        int digit = a / b;
        result += to_string(digit);
        a = a % b;
    }

    cout << result << endl;
}

int main() {
    int a, b, N;
    while (cin >> a >> b >> N) {
        divide(a, b, N);
    }
    return 0;
}
```

15. a010. 因數分解

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

int main() {
    int num;
    while (cin >> num) {
        bool first = true; // 用來判斷是否為第一個輸出項目，避免前面多印 " * "

        // 開始從 2 開始試除，直到 i*i > num 為止（因為更大的因數會配對在前面出現過了）
        for (int i = 2; i * i <= num; ++i) {
            int count = 0; // 計算質因數 i 的次方數
            while (num % i == 0) {
                num /= i;    // 不斷將 num 除以 i
                count++;     // 計數次方
            }

            // 若 i 是一個有效的質因數
            if (count > 0) {
                if (!first) cout << " * "; // 如果不是第一項，就先輸出乘號
                first = false; // 之後的項目就不再是第一項了

                // 輸出格式：
                if (count == 1) cout << i; // 若次方為 1，就只輸出 i
                else cout << i << "^" << count; // 否則輸出 i 的次方格式
            }
        }
    }
}
```

```

// 如果最後剩下的 num > 1，代表它本身也是一個質因數
if (num > 1) {
    if (!first) cout << " * "; // 若不是第一項，先補乘號
    cout << num;           // 輸出這個質因數
}
cout << endl; // 換行，開始處理下一筆輸入
}
return 0;
}

```

16. a013. 羅馬數字

題目說明：把羅馬數字轉十進位整數，算出兩個數字差的絕對值，再把答案轉回羅馬數字

```

#include <iostream>
#include <string>
#include <map>
#include <cmath>
using namespace std;

// 羅馬字串轉整數
int romanToInt(const string& s) {
    map<char, int> roman = {
        {'I', 1}, {'V', 5}, {'X', 10},
        {'L', 50}, {'C', 100}, {'D', 500}, {'M', 1000}
    };
}

```

```
};

int total = 0, prev = 0;
for (int i = s.length() - 1; i >= 0; --i) {
    int val = roman[s[i]];
    if (val < prev) total -= val;
    else {
        total += val;
        prev = val;
    }
}
return total;
}
```

// 整數轉羅馬字串

```
string intToRoman(int num) {
    int values[] = {
        1000, 900, 500, 400,
        100, 90, 50, 40,
        10, 9, 5, 4, 1
    };
    string symbols[] = {
        "M", "CM", "D", "CD",
        "C", "XC", "L", "XL",
        "X", "IX", "V", "IV", "I"
    };
    string result;
```



```
    for (int i = 0; i < 13; ++i) {
        while (num >= values[i]) {
            result += symbols[i];
            num -= values[i];
        }
    }
    return result;
}

int main() {
    string a, b;
    while (cin >> a >> b) {
        int diff = abs(romanToInt(a) - romanToInt(b));
        if (diff == 0)
            cout << "ZERO" << endl;
        else
            cout << intToRoman(diff) << endl;
    }
    return 0;
}
```

17. a528. 大數排序

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

```
using namespace std;

// 自訂比較函式：用來比較兩個大整數的字串 a 和 b，判斷 a 是否比 b 小
bool compare(const string &a, const string &b) {
    // 判斷兩個字串是否為負數
    bool negA = a[0] == '-';
    bool negB = b[0] == '-';

    // 如果 a 是負數、b 是正數，那 a < b → 回傳 true
    if (negA && !negB) return true;

    // 如果 a 是正數、b 是負數，那 a > b → 回傳 false
    if (!negA && negB) return false;

    // 去掉負號，只保留數字部分來比較大小
    string aa = negA ? a.substr(1) : a;
    string bb = negB ? b.substr(1) : b;

    // 如果數字長度不同（例如 "123" vs "4567"）
    if (aa.length() != bb.length()) {
        // 如果是負數：長度越長代表數值越小 → 越後面
        // 如果是正數：長度越長代表數值越大 → 越後面
        return negA ? (aa.length() > bb.length()) : (aa.length() < bb.length());
    }

    // 長度相同 → 比較字典序（逐字比較）
    // 若是負數，要反過來比大小（數字越大 → 實際越小）
```

```
    return negA ? (aa > bb) : (aa < bb);
}

int main() {
    int N;

    // 讀取每一組測資，直到輸入結束 (EOF)
    while (cin >> N) {
        vector<string> nums(N); // 儲存每組的大整數 (用字串表示)

        // 讀取 N 個數字
        for (int i = 0; i < N; ++i) {
            cin >> nums[i];
        }

        // 使用自訂的 compare 函式對字串進行排序 (從小到大)
        sort(nums.begin(), nums.end(), compare);

        // 輸出排序後的結果
        for (const string &s : nums) {
            cout << s << "\n";
        }
    }
    return 0;
}
```