

綜合範例

資訊之芽語法班 2014 suhorng

更多栗子 - 質數判定

- 輸入正整數 p ($2 \leq p \leq 5000$), 請輸出 p 是不是質數
("prime"/"composite")

更多栗子 - 質數判定

- 輸入正整數 p ($2 \leq p \leq 5000$), 請輸出 p 是不是質數 ("prime"/"composite")
- 分析問題:

更多栗子 - 質數判定

- 輸入正整數 p ($2 \leq p \leq 5000$), 請輸出 p 是不是質數 ("prime"/"composite")
- 分析問題:
 - p 是質數若 $2, \dots, p-1$ 都不是 p 的因數

更多栗子 - 質數判定

- 輸入正整數 p ($2 \leq p \leq 5000$), 請輸出 p 是不是質數 ("prime"/"composite")
- 分析問題:
 - p 是質數若 $2, \dots, p-1$ 都不是 p 的因數
 - 判斷是不是因數: `p%i == 0` (" p 除以 i 的餘數等於 0?")

更多栗子 - 質數判定

- 輸入正整數 p ($2 \leq p \leq 5000$), 請輸出 p 是不是質數 ("prime"/"composite")
- 分析問題:
 - p 是質數若 $2, \dots, p-1$ 都不是 p 的因數
 - 判斷是不是因數: `p%i == 0` (" p 除以 i 的餘數等於 0?")
 - 需要用迴圈跑過所有 $2 \leq i \leq p$

更多栗子 - 質數判定

- 輸入正整數 p ($2 \leq p \leq 5000$), 請輸出 p 是不是質數 ("prime"/"composite")
- 分析問題:
 - p 是質數若 $2, \dots, p-1$ 都不是 p 的因數
 - 判斷是不是因數: `p%i == 0` (" p 除以 i 的餘數等於 0?")
 - 需要用迴圈跑過所有 $2 \leq i \leq p$
 - 用一個變數 `has_factor` 紀錄 "當迴圈跑到目前為止, 是否找到任何因數?"
 - 若迴圈中 `p%i == 0`, 就把 `has_factor` 設成 1

更多栗子 - 質數判定

參考程式

更多栗子 - 質數判定

參考程式

```
#include <iostream>

int main() {
    int p;
    std::cin >> p;

    int has_factor = 0;
    for (int i = 2; i < p; i = i+1) {
        if (p%i == 0) {
            has_factor = 1;
        }
    }

    if (has_factor == 0) {
        std::cout << "prime\n";
    } else {
        std::cout << "composite\n";
    }
    return 0;
}
```

更多栗子 - 找出所有質數

- 輸入正整數 n ($2 \leq n \leq 5000$), 請輸出 $2, \dots, n$ 所有的質數

更多栗子 - 找出所有質數

- 輸入正整數 n ($2 \leq n \leq 5000$), 請輸出 $2, \dots, n$ 所有的質數
- 分析問題:

更多栗子 - 找出所有質數

- 輸入正整數 n ($2 \leq n \leq 5000$), 請輸出 $2, \dots, n$ 所有的質數
- 分析問題:
 - 我們已經會: 給一個 p , 判斷它是不是質數

更多栗子 - 找出所有質數

- 輸入正整數 n ($2 \leq n \leq 5000$), 請輸出 $2, \dots, n$ 所有的質數
- 分析問題:
 - 我們已經會: 給一個 p , 判斷它是不是質數
 - 輸出 $2, \dots, n$ 的所有質數: 把數字一一試過就好.

更多栗子 - 找出所有質數

- 輸入正整數 n ($2 \leq n \leq 5000$), 請輸出 $2, \dots, n$ 所有的質數
- 分析問題:
 - 我們已經會: 給一個 p , 判斷它是不是質數
 - 輸出 $2, \dots, n$ 的所有質數: 把數字一一試過就好.
 - 用一個迴圈 i 跑過 $2 \leq i \leq n$, 每次迴圈中, 若 i 是質數, 就印出 i
 - 又是一個**巢狀迴圈**的應用

更多栗子 - 找出所有質數

參考程式

更多栗子 - 找出所有質數

參考程式

```
#include <iostream>

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

    for (int i = 2; i <= n; i = i+1) {
        "計算 i 是不是質數"

        if (has_factor == 0) {
            std::cout << i << "\n";
        }
    }
    return 0;
}
```


更多栗子 - 找出所有質數

參考程式

```
#include <iostream>

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

    for (int i = 2; i <= n; i = i+1) {
        int p = i;
        int has_factor = 0;
        for (int i = 2; i < p; i = i+1) {
            if (p%i == 0) {
                has_factor = 1;
            }
        }

        if (has_factor == 0) {
            std::cout << i << "\n";
        }
    }
    return 0;
}
```

- 問題: 可以改掉 `int p = i;` 這種 "重新命名" 的動作嗎?

更多栗子 - 找出所有質數

- 但是 i 撞名了!
- 為何目前為止兩個 i 不會出事?

```
#include <iostream>

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

    for (int i = 2; i <= n; i = i+1) {
        int p = i;
        int has_factor = 0;
        for (int i = 2; i < p; i = i+1) {
            if (p%i == 0) {
                has_factor = 1;
            }
        }

        if (has_factor == 0) {
            std::cout << i << "\n";
        }
    }
    return 0;
}
```

更多栗子 - 找出所有質數

- 回憶: **for** 投影片講的 **scope**
- 也是可以改名, 像下面這樣

```
#include <iostream>

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

    for (int i = 2; i <= n; i = i+1) {
        int has_factor = 0;
        for (int j = 2; j < i; j = j+1) {
            if (i%j == 0) {
                has_factor = 1;
            }
        }

        if (has_factor == 0) {
            std::cout << i << "\n";
        }
    }
    return 0;
}
```

加分作業 -- 樹大招風 (共 3 小題)

- 輸入第一行是正整數 n ($1 \leq n \leq 500$), 接著
- 輸入 n 個正整數 a_0, \dots, a_{n-1} , 重複以下的動作 n 次, 每次時
 - 把當前的最大值都設成 -1
 - 印出一行當前的 a_0, \dots, a_{n-1} , 數字間用一個空白隔開
- 例如輸入

```
4
5 3 5 1
```

- 則輸出: (每行行末不要輸出空白, 最後一行要換行)

```
-1 3 -1 1
-1 -1 -1 1
-1 -1 -1 -1
-1 -1 -1 -1
```

加分作業 -- 樹大招風 第 1 小題

- 輸入第一行是正整數 n ($1 \leq n \leq 500$), 接著
- 輸入 n 個正整數 a_0, \dots, a_{n-1} , 請輸出它們的最大值
- 例如輸入

```
4  
5 3 5 1
```

- 則輸出: (要換行)

```
5
```

加分作業 -- 樹大招風 第 2 小題

- 輸入第一行是正整數 n ($1 \leq n \leq 500$), 接著
- 輸入 n 個正整數 a_0, \dots, a_{n-1} , 把最大值設成 -1 後輸出
- 例如輸入

```
4  
5 3 5 1
```

- 則輸出: (數字間以一個空白隔開, 結尾不要空白, 請換行)

```
-1 3 -1 1
```

加分作業 -- 樹大招風 第 2 小題

- 輸入第一行是正整數 n ($1 \leq n \leq 500$), 接著
- 輸入 n 個正整數 a_0, \dots, a_{n-1} , 把最大值設成 -1 後輸出
- 例如輸入

```
4
5 3 5 1
```

- 則輸出: (數字間以一個空白隔開, 結尾不要空白, 請換行)

```
-1 3 -1 1
```

- Hint: 用上一小題找出最大值, 存在變數 m 中
- 對每個數字 a_0, \dots, a_{n-1} , 假設它等於 m , 就把它改成 -1

加分作業 -- 樹大招風 第 2 小題 (續)

- 印數字的方法:
 - 可以先印出 `a[0]`
 - 然後用迴圈印出 `a[1]` 到 `a[n-1]`, 並印著空白
 - 還有結尾的換行!!

加分作業 -- 樹大招風 第 2 小題 (續)

- 印數字的方法:
 - 可以先印出 `a[0]`
 - 然後用迴圈印出 `a[1]` 到 `a[n-1]`, 並印著空白
 - 還有結尾的換行!!

```
// 印出 a[0], ..., a[n-1]
std::cout << a[0];

for (int i = 1; i < n; i = i+1) {
    std::cout << " " << a[i];
}

std::cout << "\n";
```

加分作業 -- 樹大招風 第 3 小題

- 題目請參照第 20 頁
- Hint:

加分作業 -- 樹大招風 第 3 小題

- 題目請參照第 20 頁
- Hint:
 - 跑 n 次迴圈, 在每一次迴圈中, 做一第 2 小題
 - 小心變數名稱有沒有重複

加分作業 -- 樹大招風 第 3 小題

- 題目請參照第 20 頁
- Hint:
 - 跑 n 次迴圈, 在每一次迴圈中, 做一次第 2 小題
 - 小心變數名稱有沒有重複

```
int n;  
int a[500];  
  
"輸入 n, 然後輸入 n 個數字, 存到 a 中"  
  
for (int round = 0; round < n; round = round+1) {  
    int m = -1;  
  
    "找最大值, 並存到 m 中"  
  
    "把 a 裡面所有 == m 的數字都改成 -1"  
  
    "印出 a[0], ..., a[n-1]"  
}
```