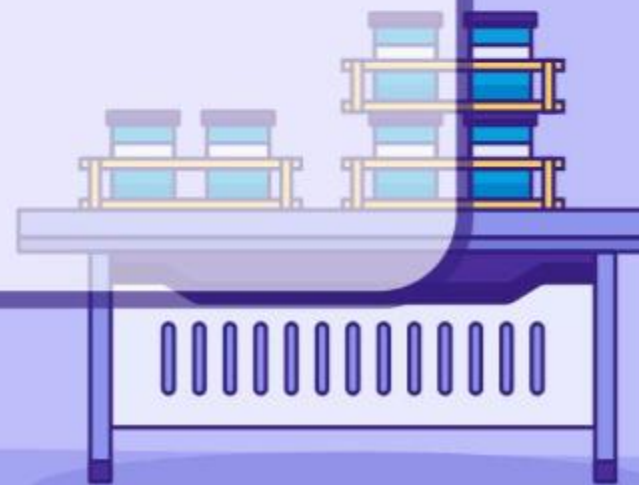
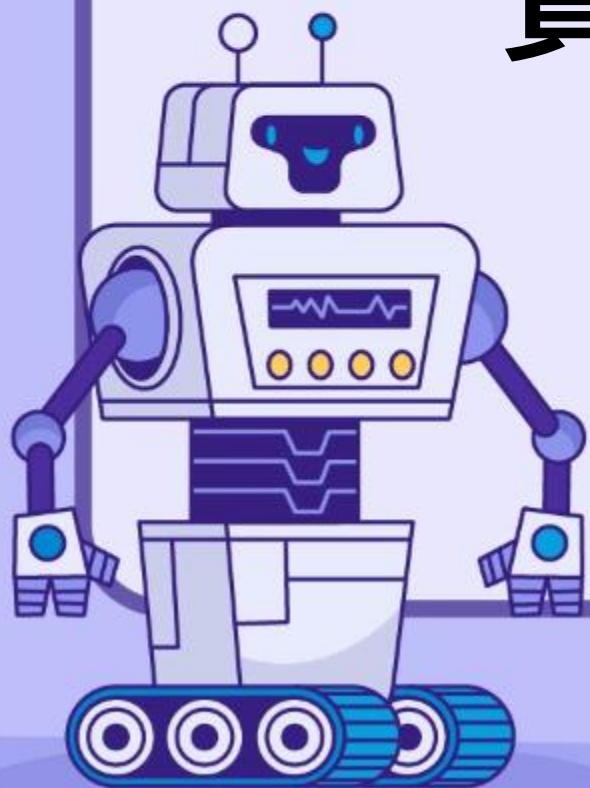


Younger DreamworkShop

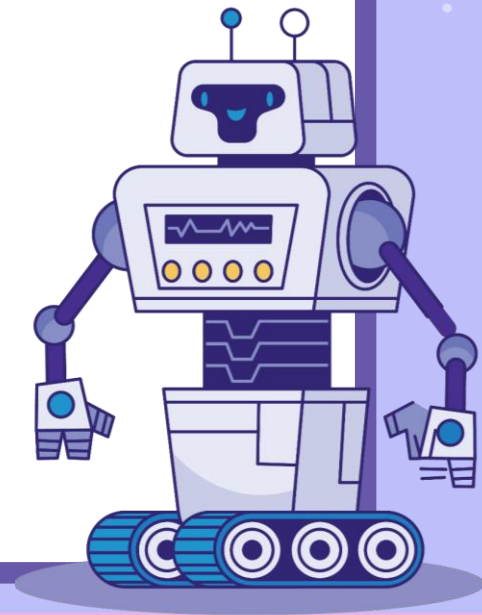
資訊科技競賽培訓教材

北興科技中心楊心淵



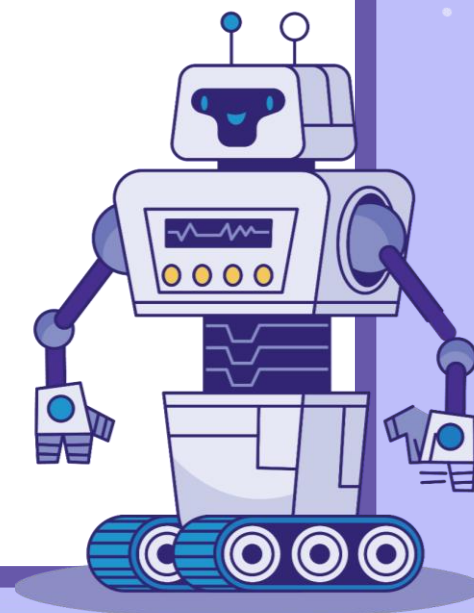
選手培訓推動策略共構

- 縣市普及、全國賽菁英
- Scratch、Block雙軌自選
- 融入學校課程(Scratch選拔選手)
 - 七年級結構化程式設計
 - 八年級陣列模組化程式設計
- 師生共學(Blockly選手共訓)
 - 分階序列學習活動
 - 學習型競賽-搭配實體



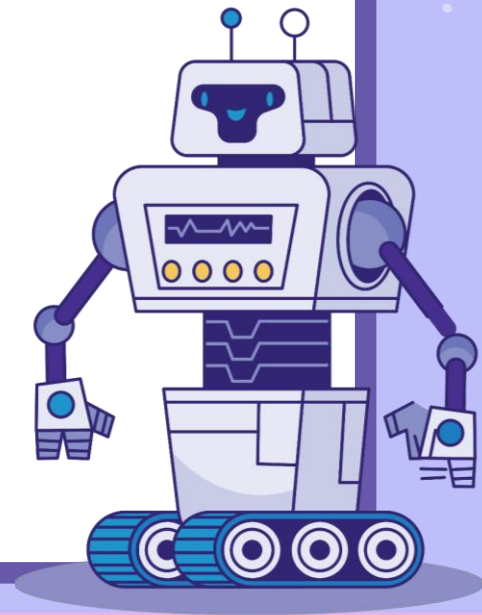
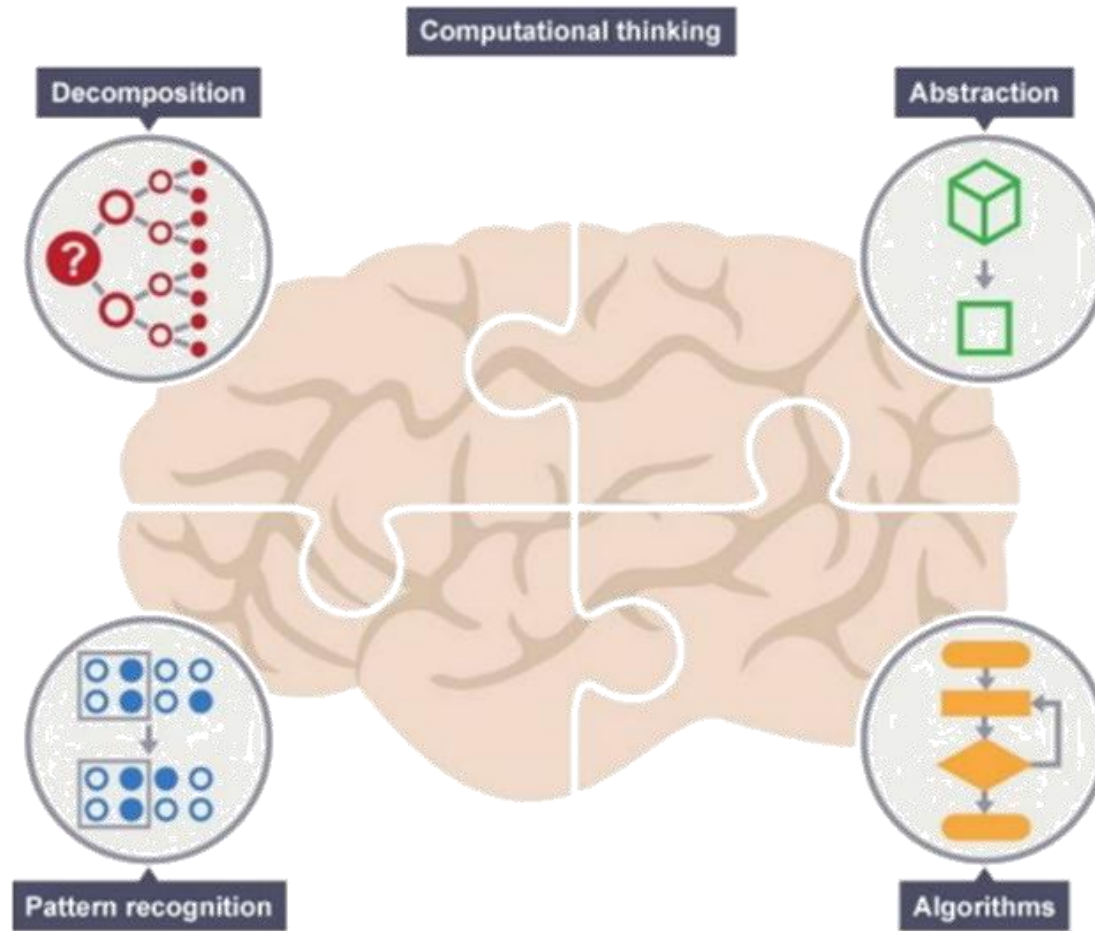
Task list

- 資訊科技課程理念
- 資訊科技競賽方向
- 資訊科技競賽解題闖通關
- 運算思維小遊戲 & Bebras 解題策略
- 問題導向程式設計初探



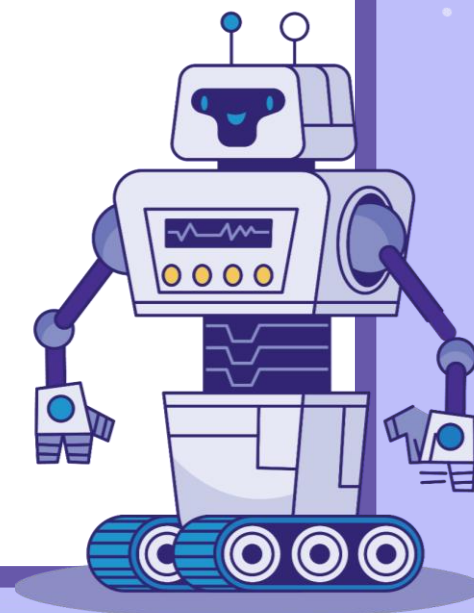
運算思維

- 拆解
- 樣式辨識
- 抽象化
- 演算法設計



抽象化

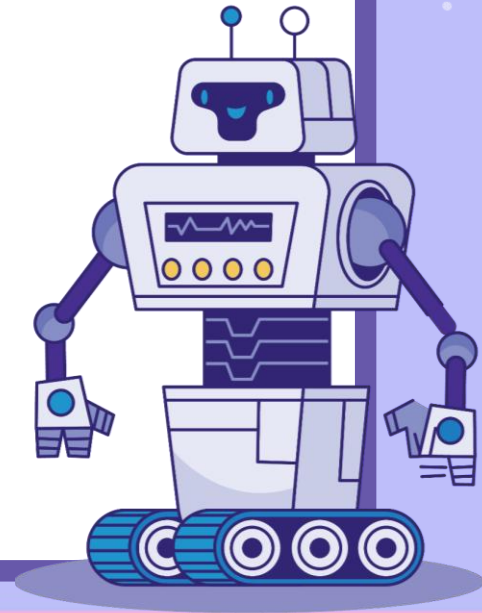
- 把生活中情境問題，用數字符號表達
- 只聚焦在問題的本質
- 捷運地圖
- 那些商店，幾個紅綠燈....
- 站與站的關聯



樣式辨識

- 在資料中觀察樣式、趨勢或規則
- 預測或簡化問題
- 產出共通的模式、規則、原則或理論

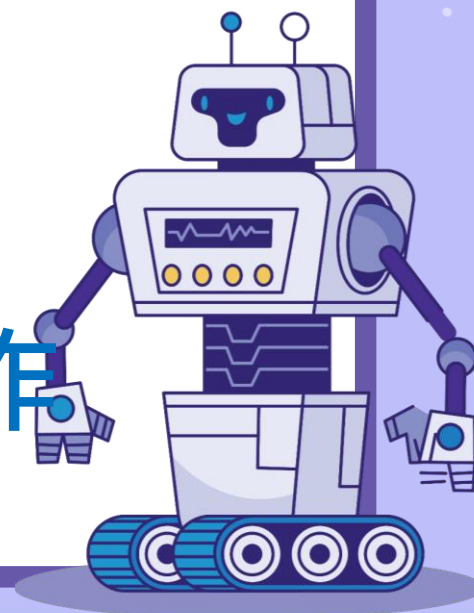
從資料找出規則
用程式表達



拆解

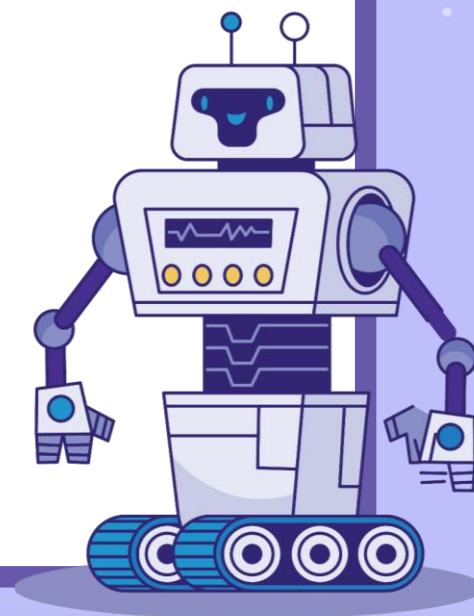
- 大問題拆解小問題，一一解決，各個擊破
- 可同時處理各個小問題
- 最後仍要整合

平行處理、專業分工、密切合作



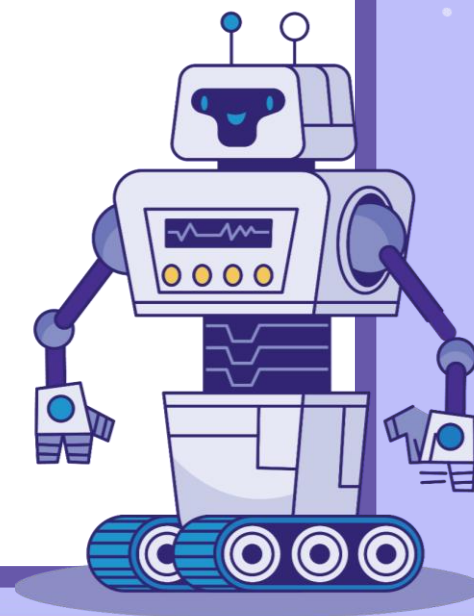
演算法

- 將情境問題抽象化(數字或符號表達)
- 找出規則規律，以程式表達
 - 輸入(注意格式)
 - 處理
 - 依題意運算分析
 - 單結構迴圈
 - 雙結構迴圈
 - 清單數據存取
 - 輸出

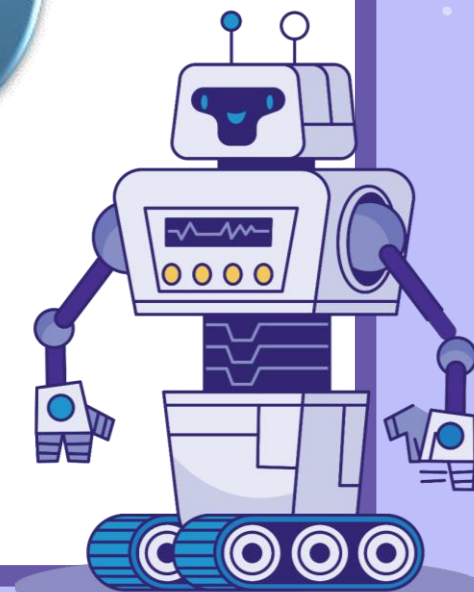
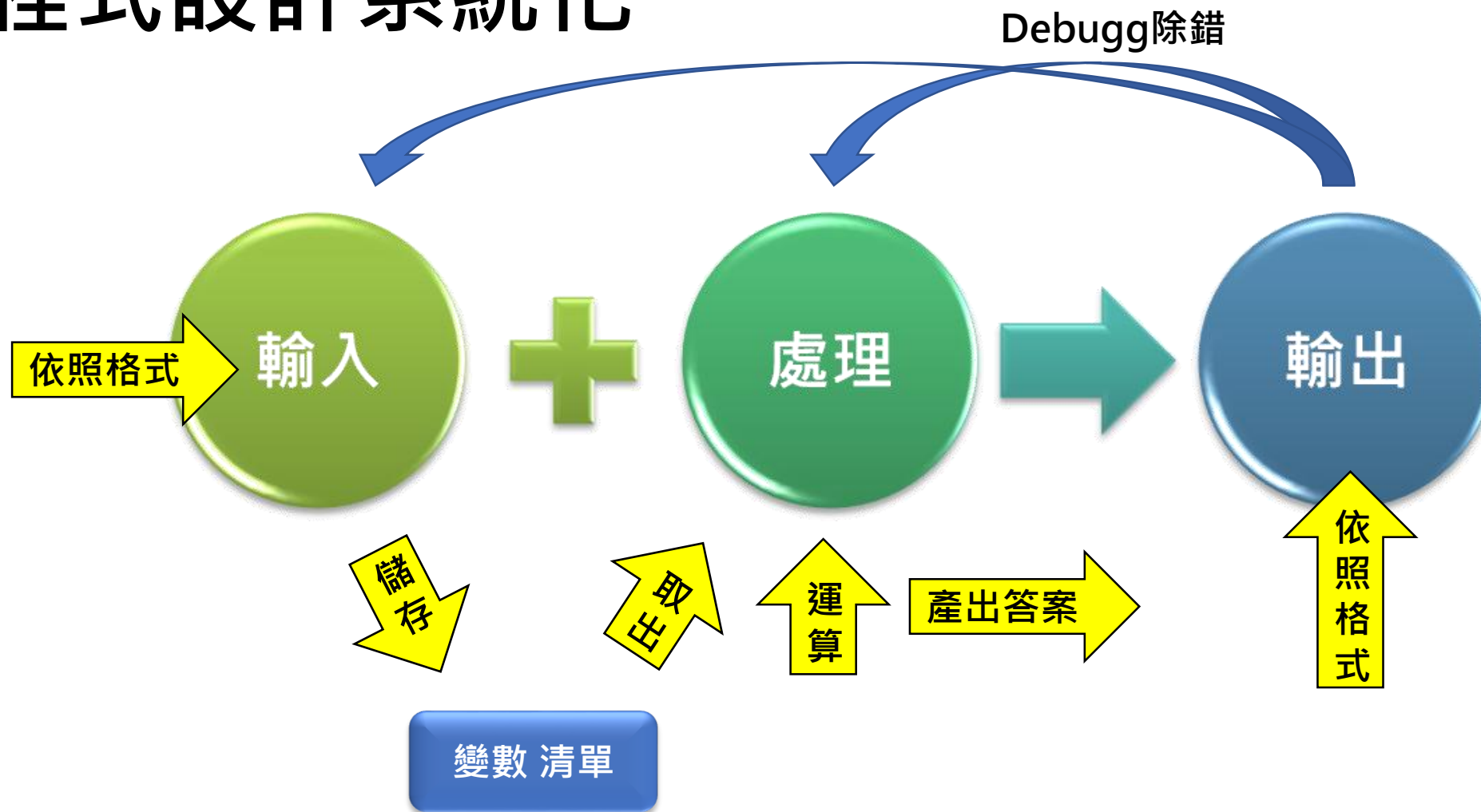


競賽方向

- 個人賽：
 - 程式設計平台介面熟悉
 - 程式設計基本概念
 - 程式設計解題應用
- 團體賽：演算法應用
 - 演算法運用
 - 團隊合作

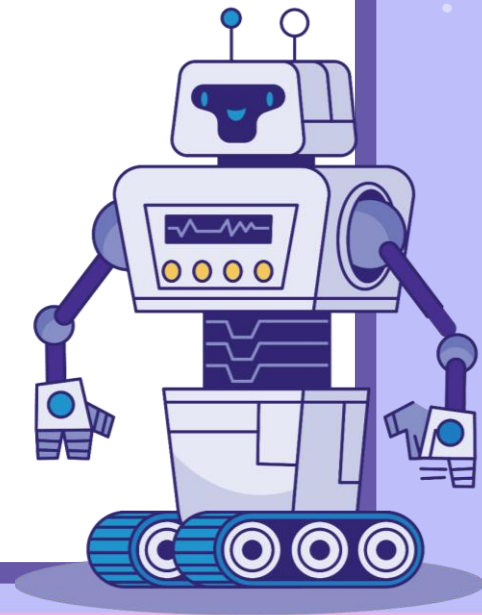


程式設計系統化



程式解題流程

- 理解題意：閱讀理解題目敘述與確認題目輸入、輸出格式
- 輸入：依照題目說明**格式**，讀取輸入資料，並**記錄至變數(清單)**
- 處理：依題意**讀取變數(清單)**資料，進行程式演算(**運算分析**)，求得答案
- 輸出：將答案依題目要求格式輸出
- 測試&除錯：理解題目敘述與題目的輸入輸出格式



課程代碼

- 基礎1：20260301S1
- 基礎2：20260301S2

YoungerDreamWorks-START

通關密語: 20260301S1

載入配置

成功載入配置!

切換模式: 學習模式



1-0-Hello world

[前往 Blockly 網站](#)

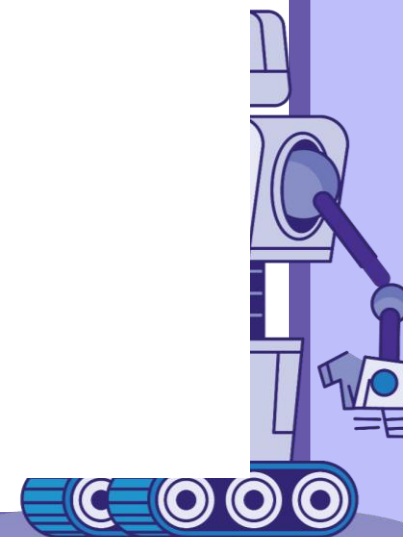


1-0-Hello world

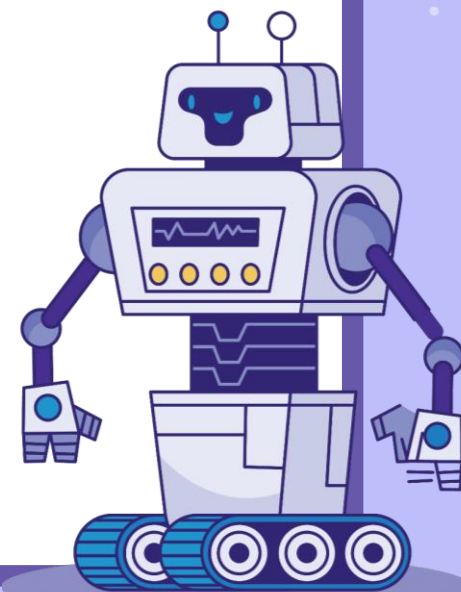
[前往 Scratch 網站](#)

點擊上方連結探索更多！

本軟體由桃園市八德國小 蔡佳倫 老師製作 Mail:crousekimo@yahoo.com.tw



1-0-Hello world



理解題意



輸入



處理



輸出

Hello world

題目說明

在程式設計的第一步，最重要的是能夠讀取使用者輸入，並將結果正確輸出。

請你寫一個小程式，讓使用者輸入自己的名字，然後程式要輸出一行「Hello, 名字」。

Hello後面會接小寫逗號，還有一個空白輸入

這樣的練習可以幫助你理解輸入與輸出，是學習程式設計的基礎。

範例格式

使用者輸入	輸出	範例說明
Amy	Hello, Amy	第一個輸入 Amy，代表名字為 Amy 程式輸出 Hello, Amy
Tom	Hello, Tom	第一個輸入 Amy，代表名字為 Amy 程式輸出 Hello, Amy

1-0

關鍵：看清題目，注意空格

輸入



處理

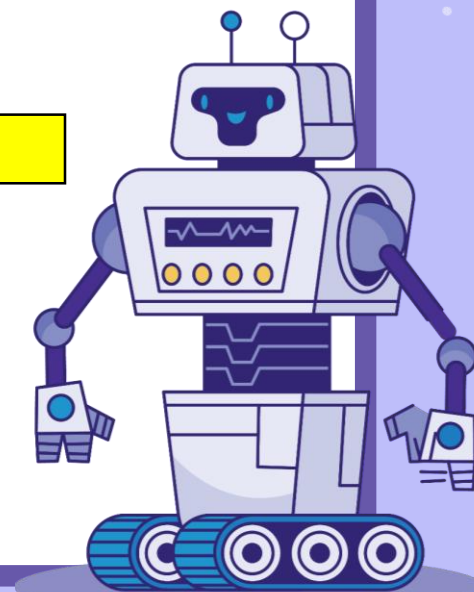


輸出



1變數用來儲存輸入的資料

2資料運算(文字相加)



輸入



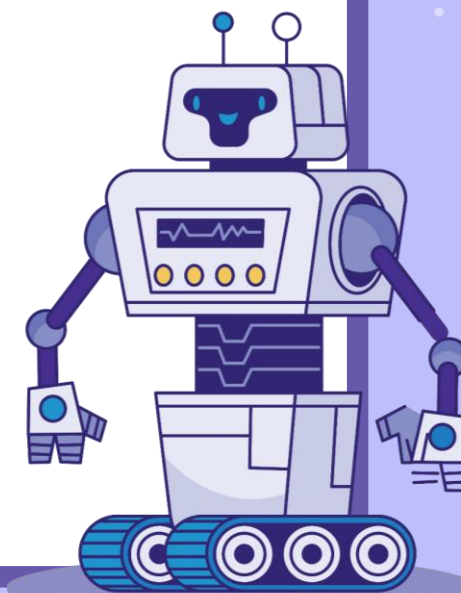
處理



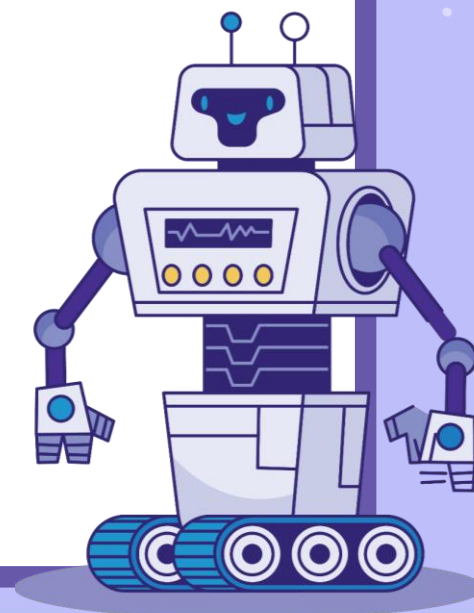
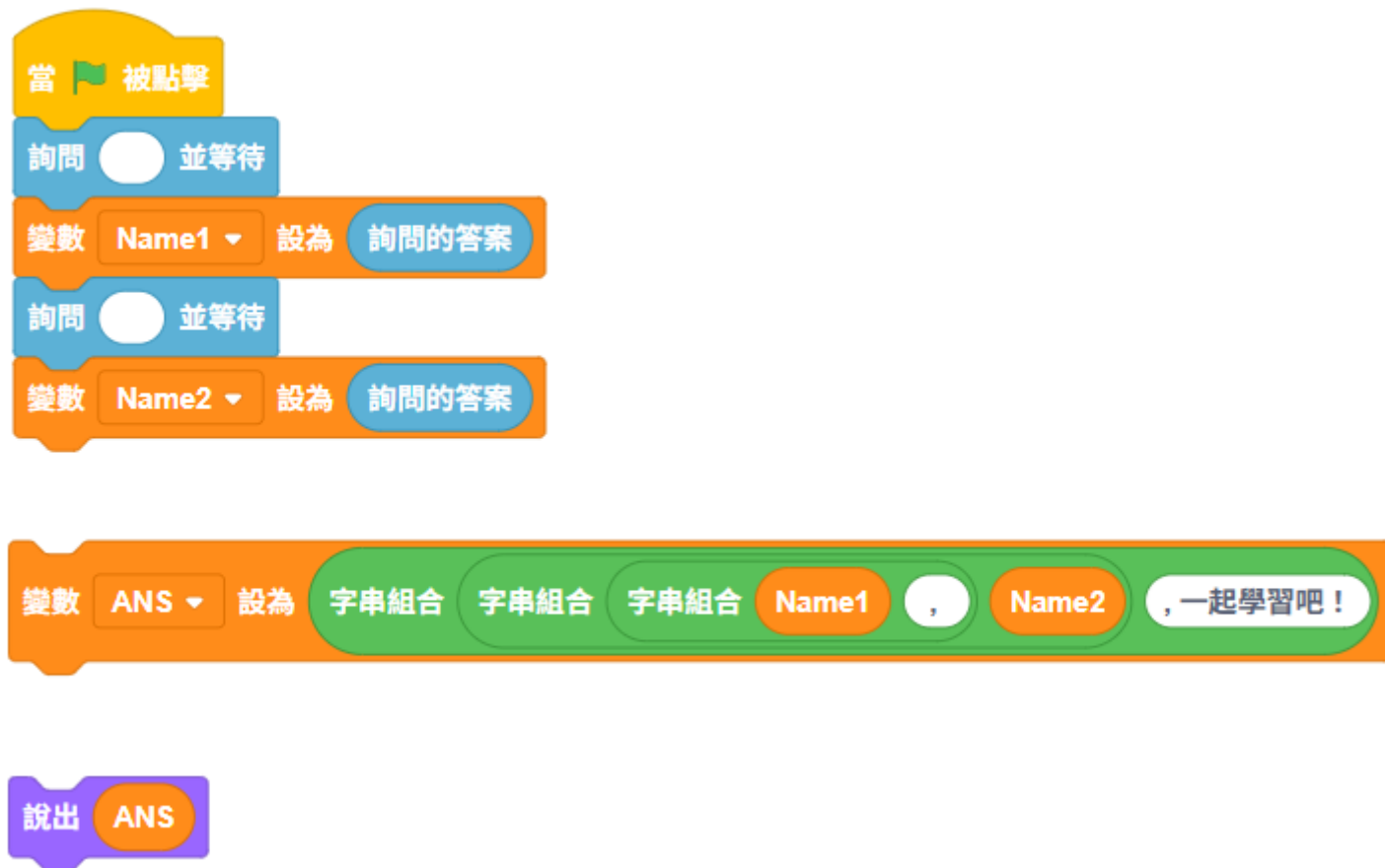
輸出

關鍵：複製題目文字作答

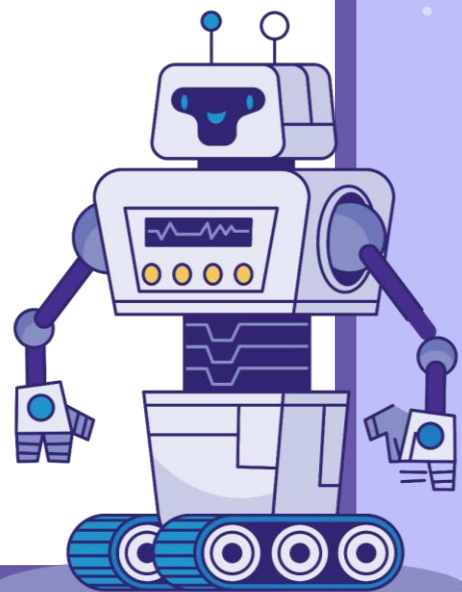
1-1 換你試試看！

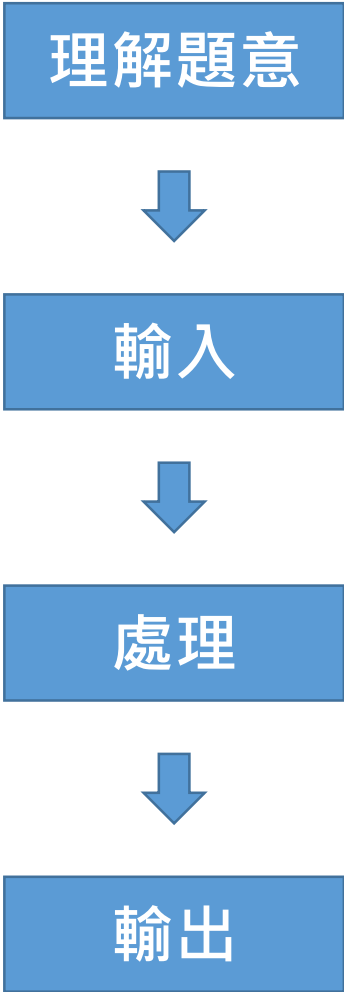


1-1



2-0-數字加總





數字加總

題目說明

當我們要處理數字時，常常需要加總。
請設計一個程式，讓使用者輸入兩個數字，程式要輸出這兩個數字的總和。
這樣的題目能幫助你熟悉數字資料處理。

範例格式

使用者輸入	輸出	範例說明
9 4	13	第一個輸入9 第二個輸入4 程式輸出9+4總和13
6 5	11	第一個輸入6 第二個輸入5 程式輸出6+5總和11

輸入



處理



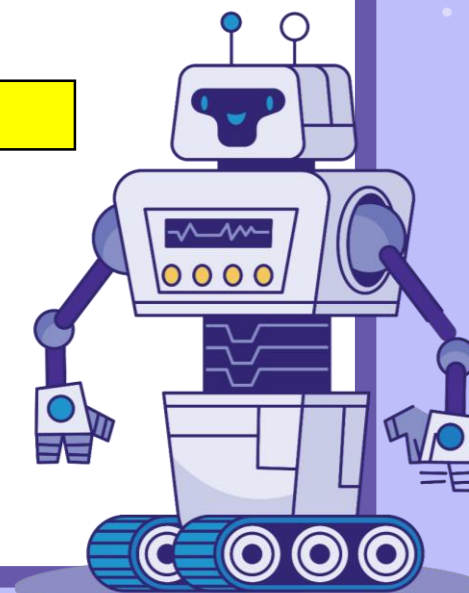
輸出



關鍵：文字、數字格式？

1變數用來儲存輸入的資料

2資料運算(數字相加)



輸入



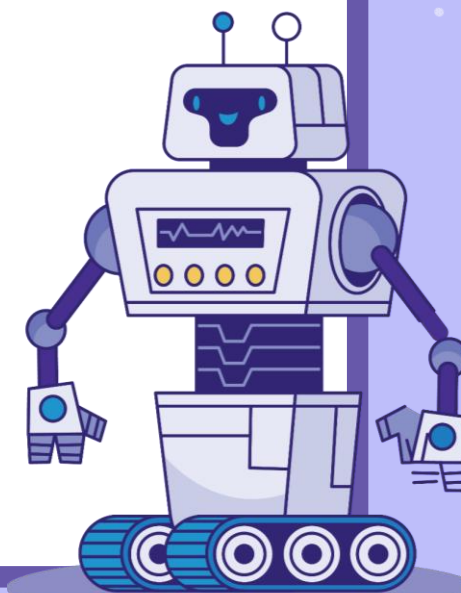
處理



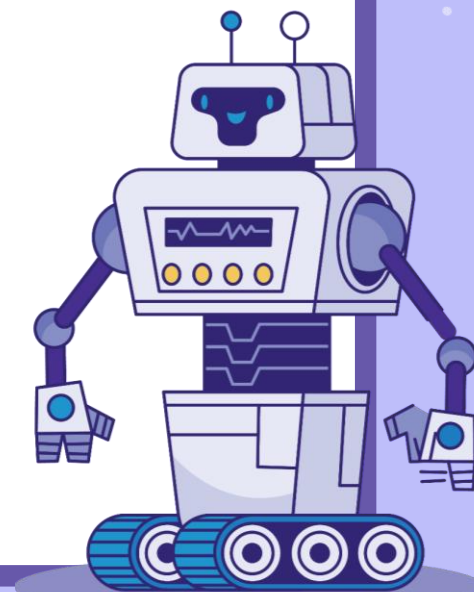
輸出

關鍵：看清題目、數字運算

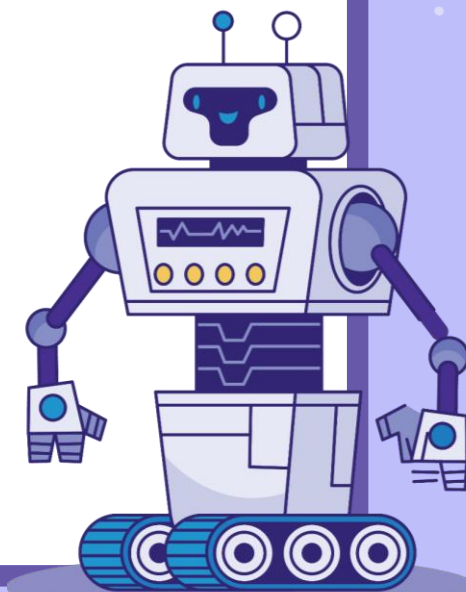
2-1 換你試試看！

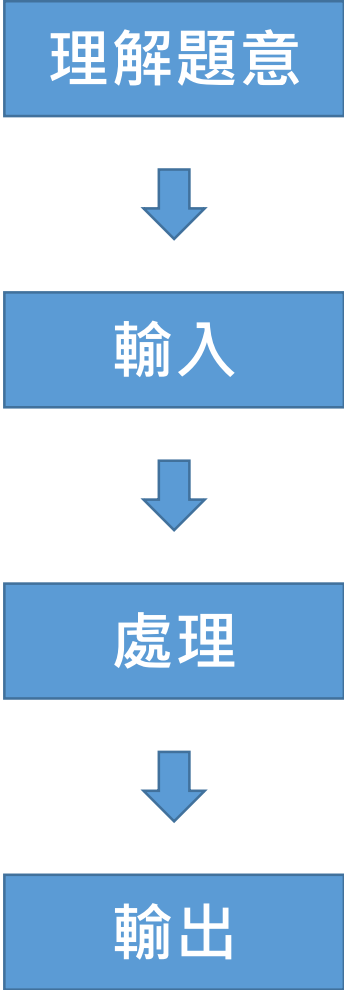


2-1



3-0-比較大小





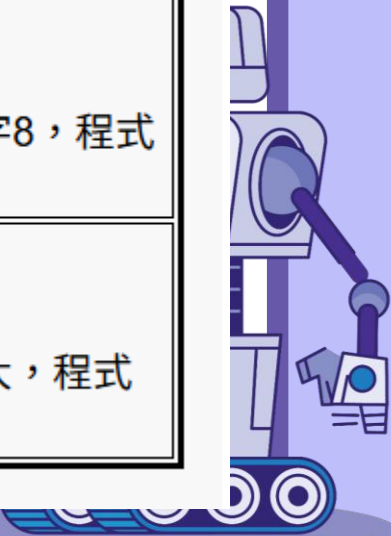
比較大小

題目說明

在生活中常常會需要比較兩個數字誰比較大。
請寫一個程式，輸入兩個整數，輸出較大的數字。
如果兩個數字相同，就輸出「一樣大」。
這樣的練習可以訓練你使用條件判斷。

範例格式

使用者輸入	輸出	範例說明
8 5	8	第一個輸入8 第二個輸入5 較大的是數字8，程式輸出8
7 7	一樣大	第一個輸入7 第二個輸入7 二個數一樣大，程式輸出一樣大



輸入



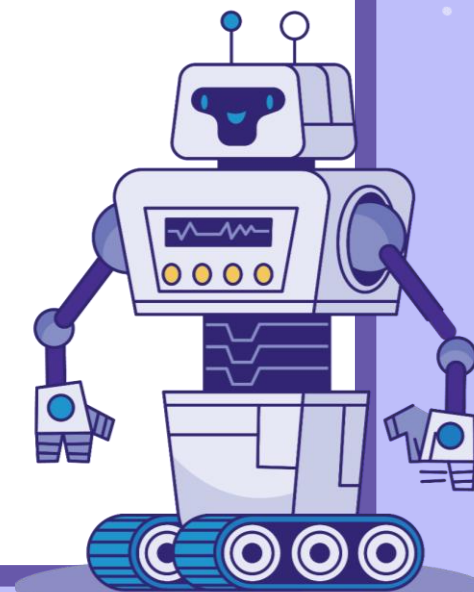
處理



輸出



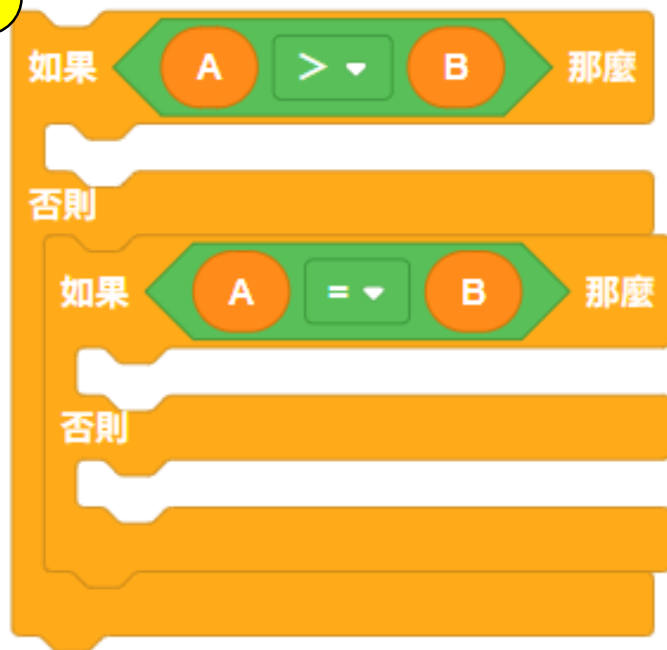
關鍵：選擇結構



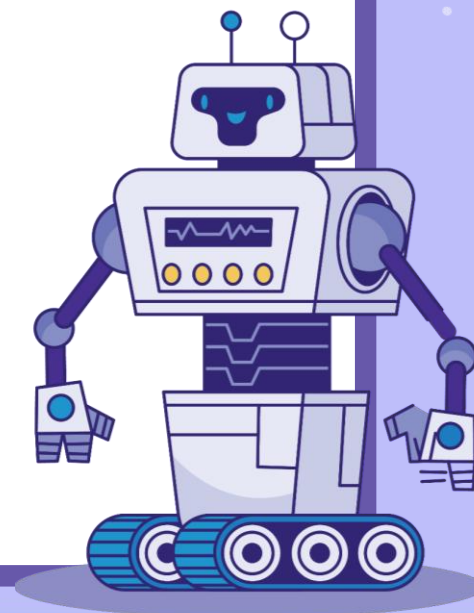
CaseStudy

想想看，那一個比較好？

1



2



輸入



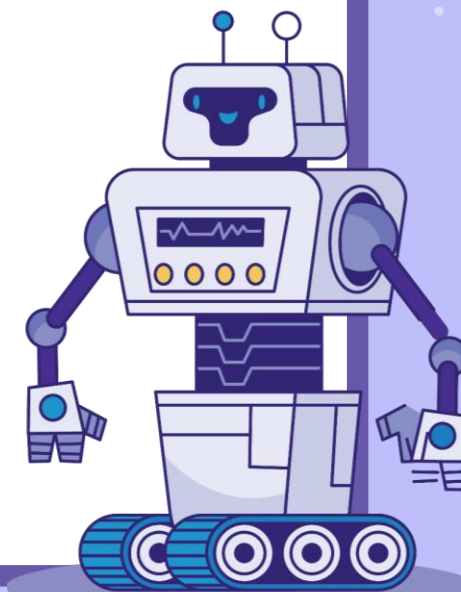
處理



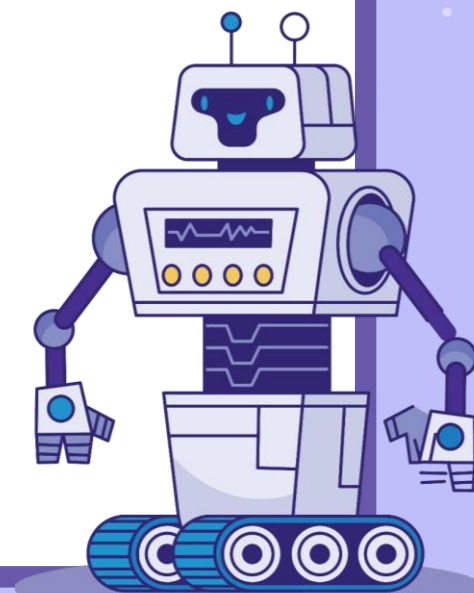
輸出

注意錯誤訊息說明來除錯

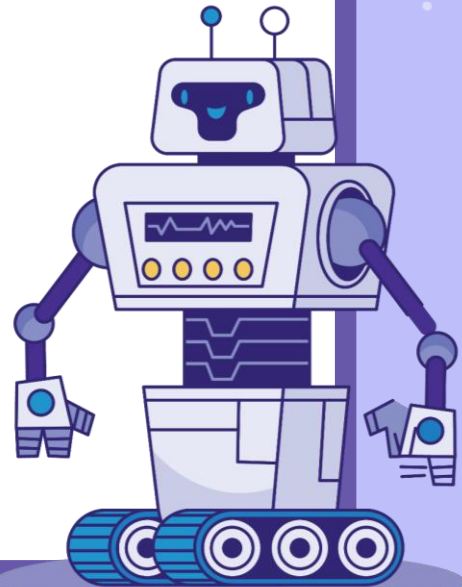
3-1 換你試試看！



3-1



4-0-判斷奇偶數



理解題意



輸入



處理



輸出

判斷奇偶數

題目說明

程式設計中，常常需要根據數字的性質進行分類。
請你寫一個程式，輸入一個整數，如果是偶數，輸出 Even，如果是奇數，輸出 Odd。
這樣的練習可以幫助你熟悉條件判斷的應用。

範例格式

使用者輸入	輸出	範例說明
8	Even	輸入8，8除以2餘數=0 程式輸出偶數Even
7	Odd	輸入7，7除以2餘數=1 程式輸出奇數數Odd

輸入



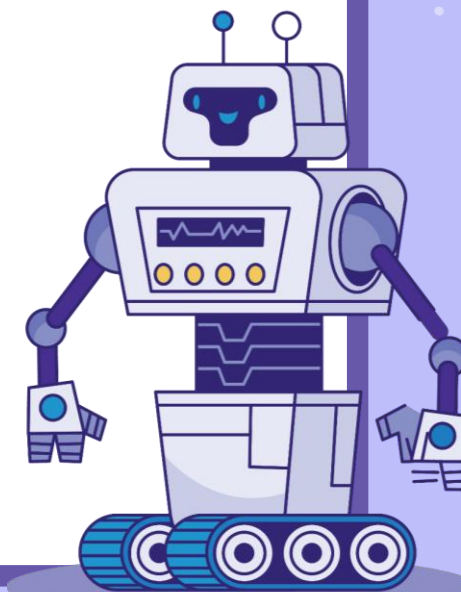
處理

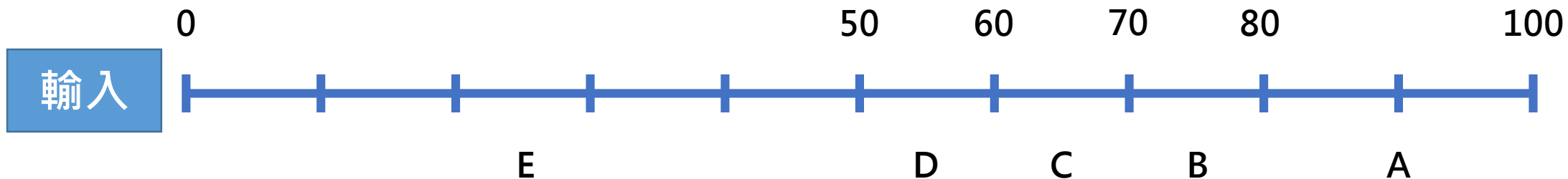


輸出



判斷奇偶數的演算法



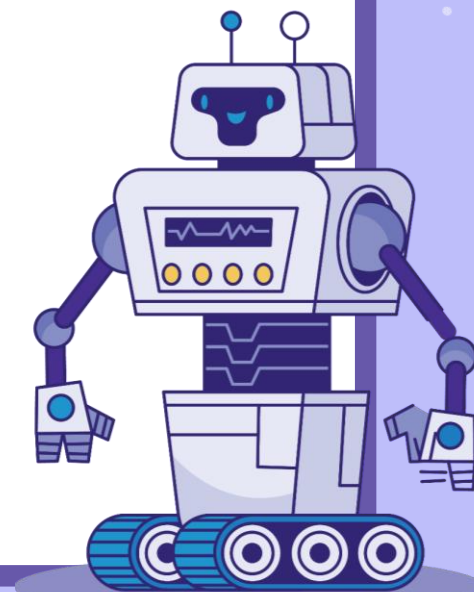


處理

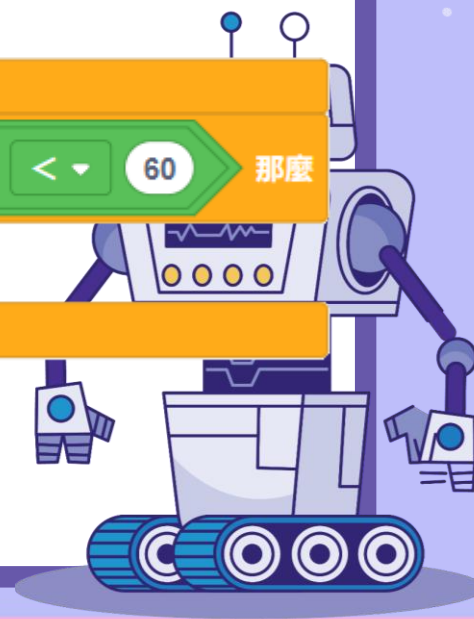
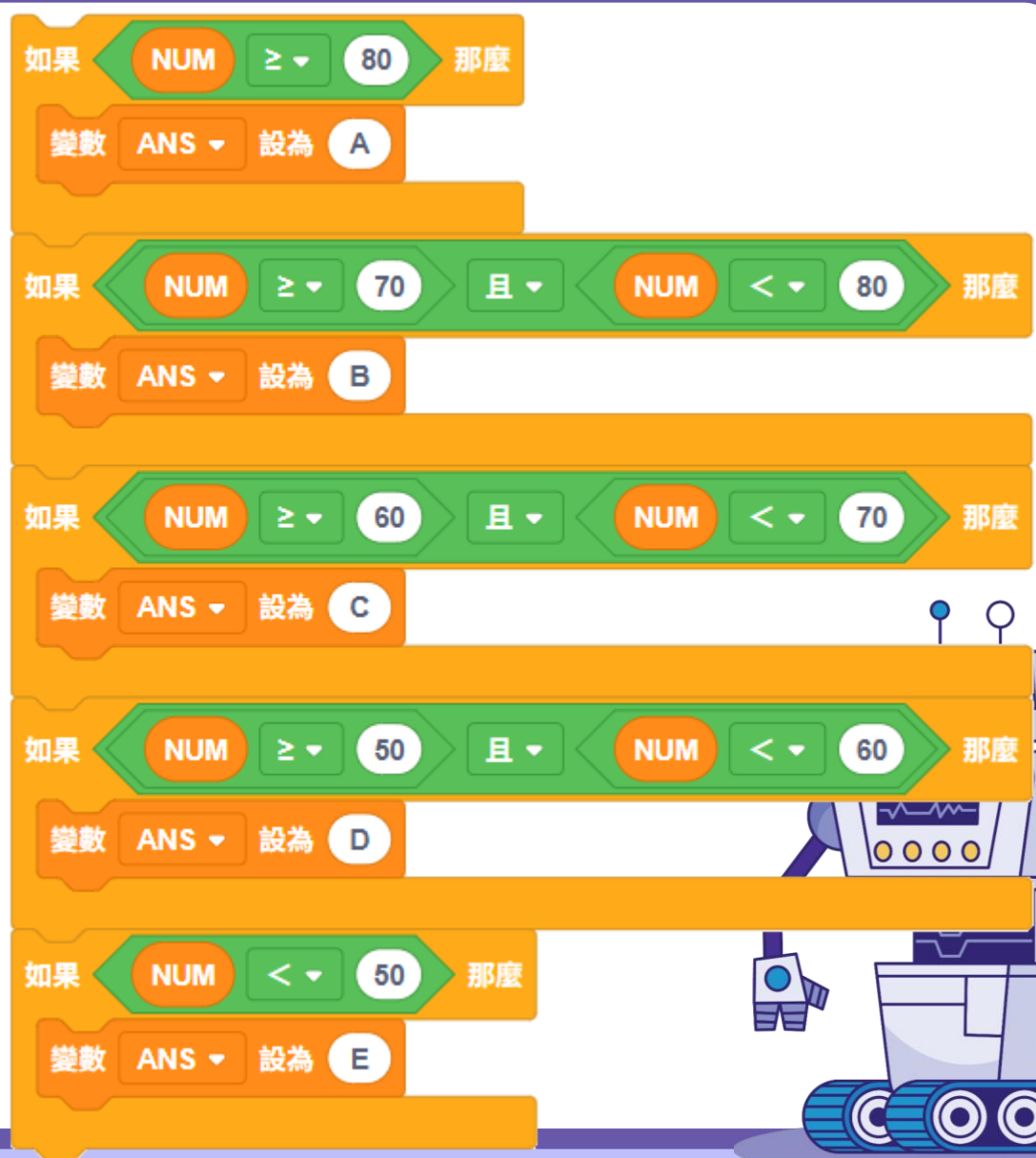


輸出

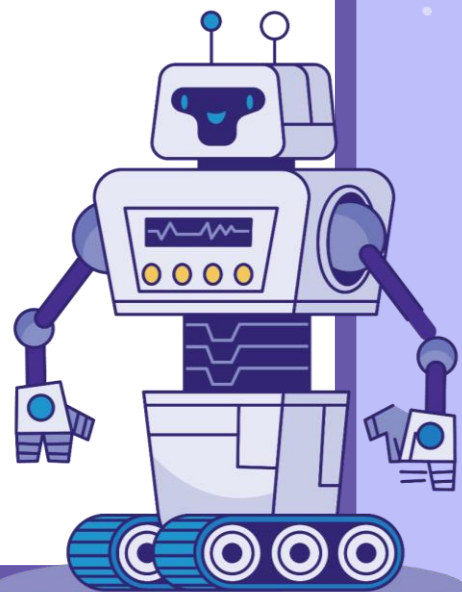
4-1 換你試試看！



4-1



5-0-三科成績計算



三科成績計算

題目說明

小明參加了數學、英文、自然三科的段考，老師希望能快速算出總分與平均分數，並判斷是否及格。

請設計程式，輸入三科的成績，計算總分與平均。

平均成績以四捨五入取到整數

如果平均分數大於等於 60，就輸出「及格」，否則輸出「不及格」。

這題可以訓練你如何處理多筆資料的加總、平均計算，並搭配條件判斷。

範例格式

使用者輸入	輸出	範例說明
3 80 70 90	240 80 及格	第一行輸入3,表示有3筆資料 第二行輸入3筆資料80 70 90，經過計算後總分、平均、及格或不及格180 80 及格
3 50 42 60	152 51 不及格	第一行輸入3,表示有3筆資料 第二行輸入3筆資料50 42 60，經過計算後程式輸出總分、平均、及格或不及格152 51 不及格

理解題意



輸入



處理



輸出

輸入



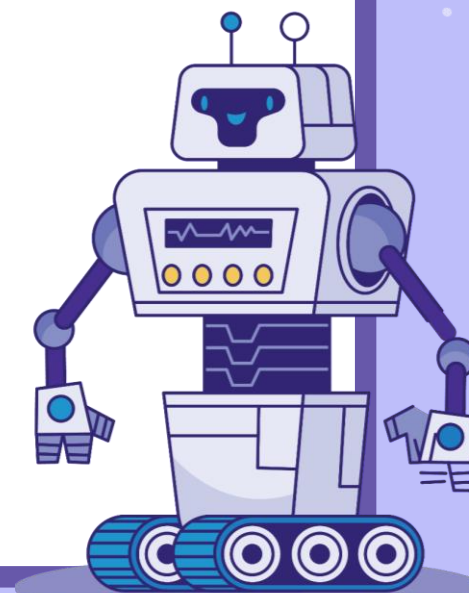
處理



輸出



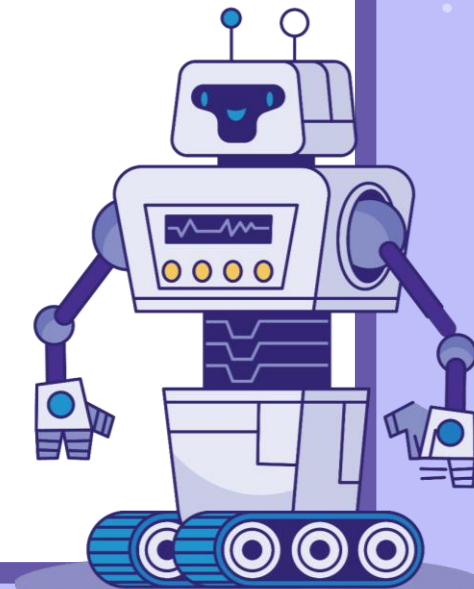
批次輸入



Case Study 輸出格式



輸出
240
80
及格
152
51
不及格



輸入

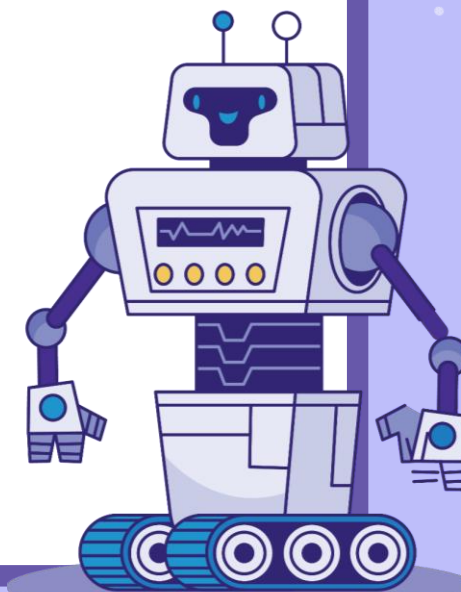


處理

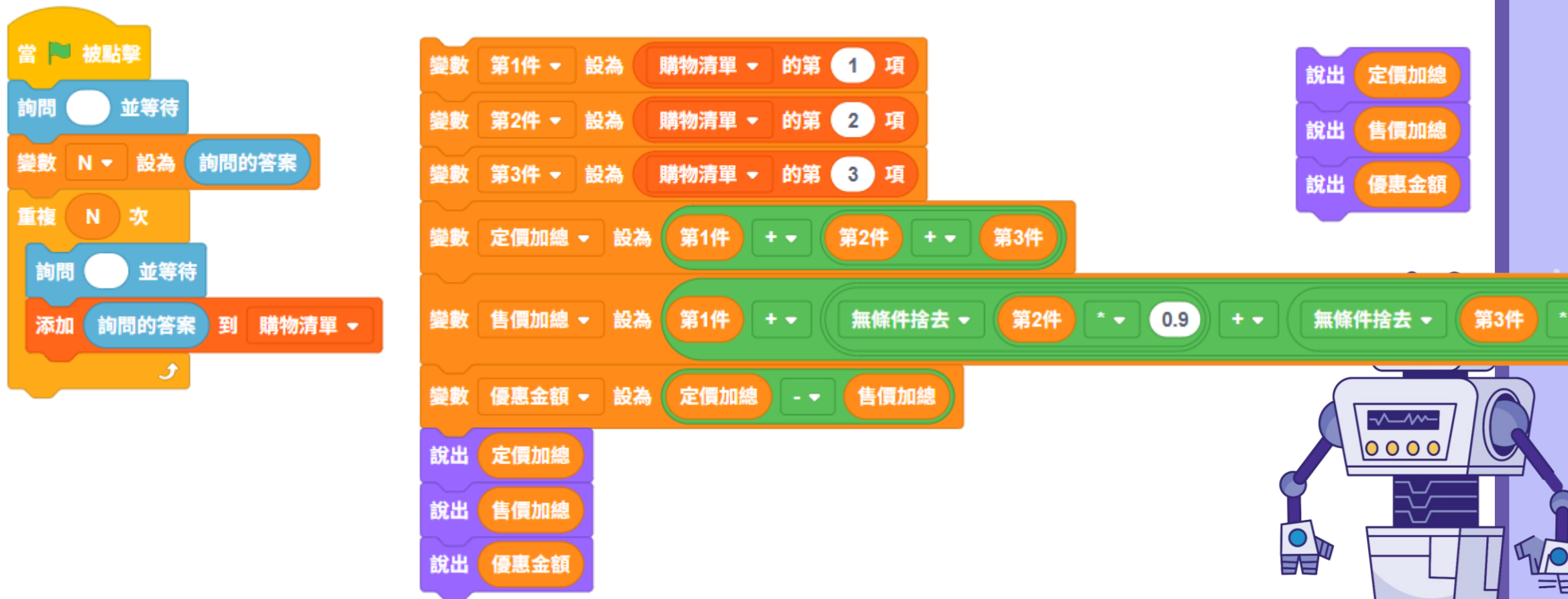


輸出

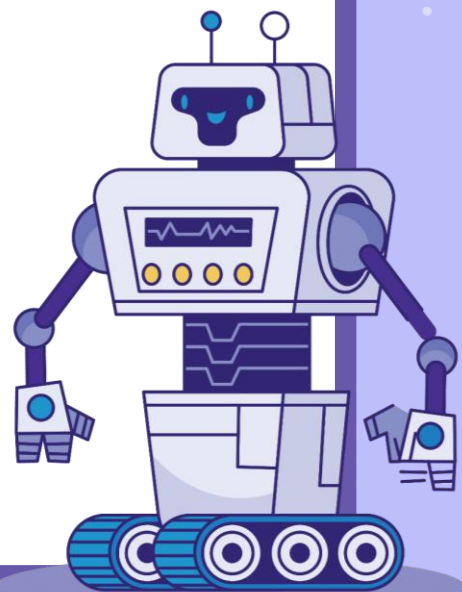
5-1 換你試試看！

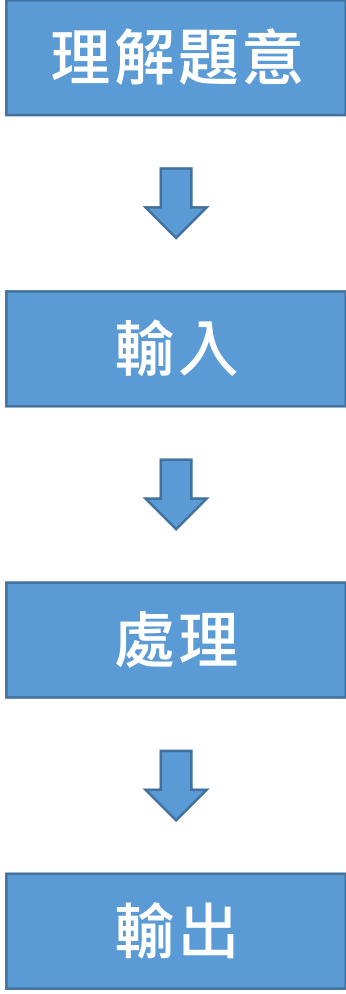


5-1



6-0-連續加總





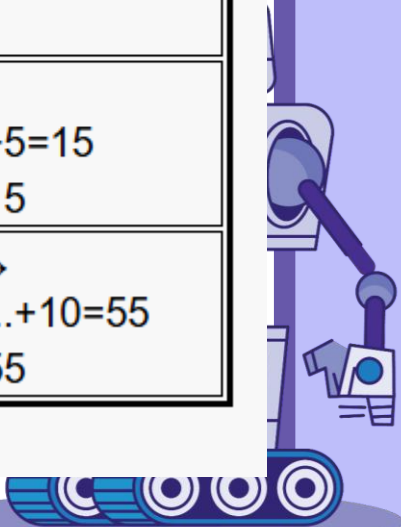
連續加總

題目說明

有時候我們需要計算從 1 到某個數字的總和。
請寫一個程式，輸入一個整數 N，計算從 1 加到 N 的結果。
例如 N=5 時， $1+2+3+4+5=15$ 。這樣的練習可以訓練你使用迴圈進行加總。

範例格式

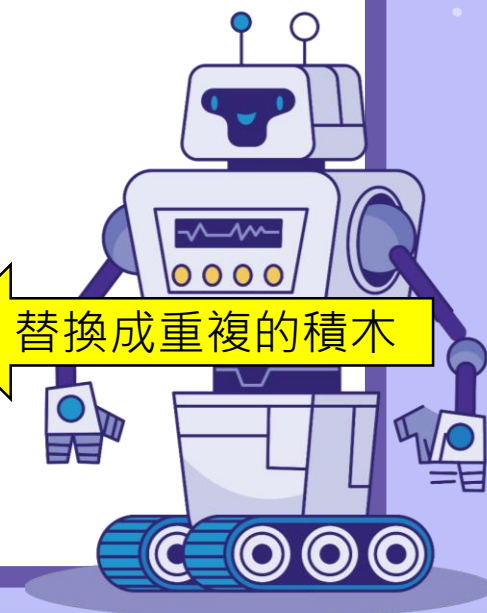
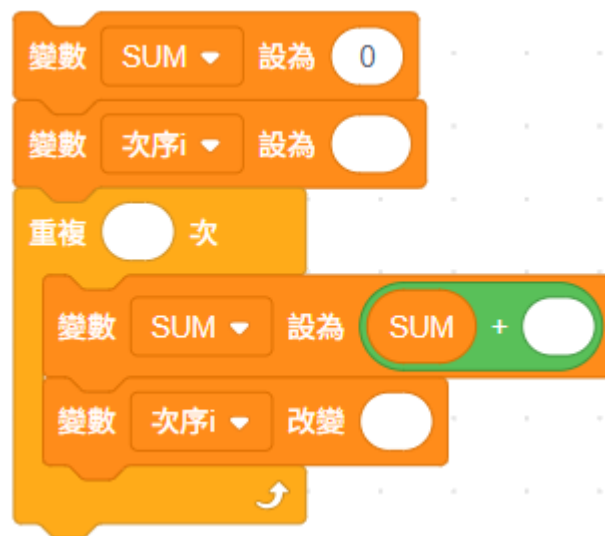
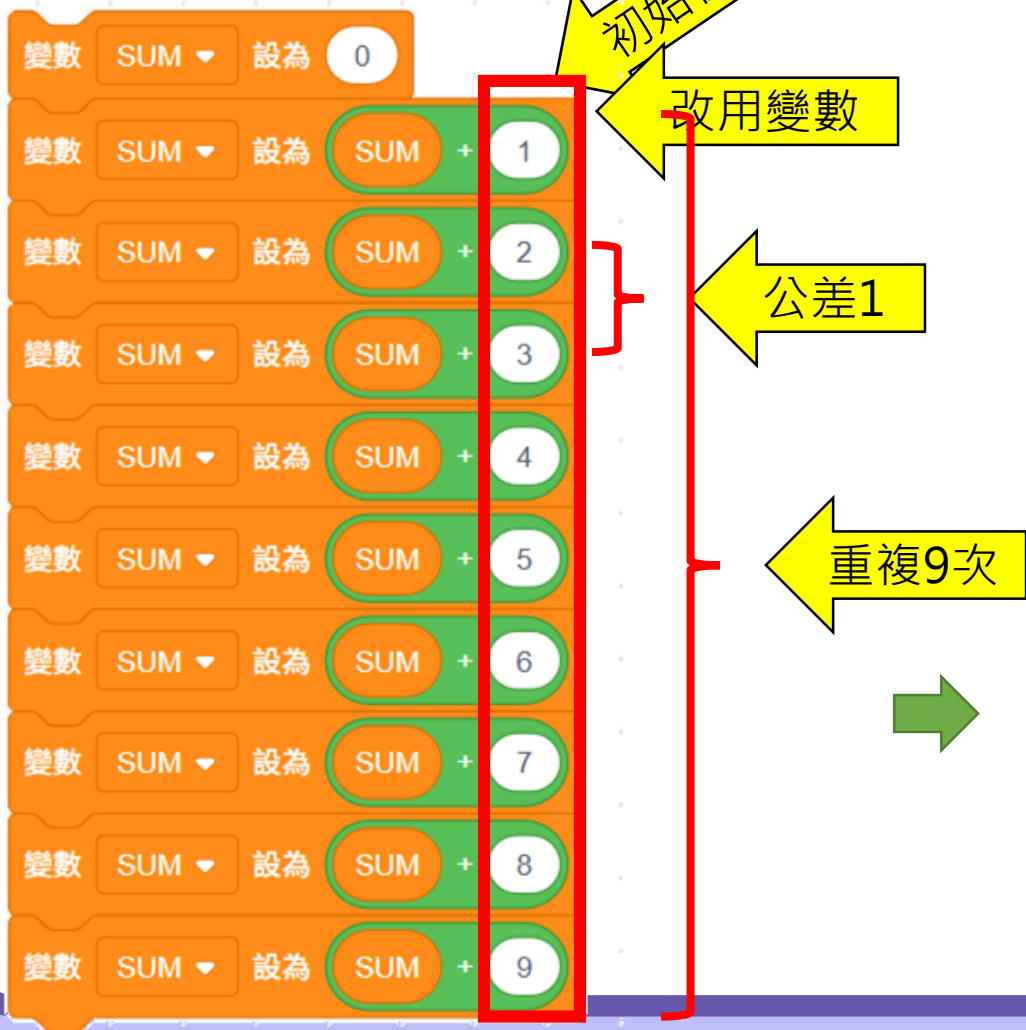
使用者輸入	輸出	範例說明
5	15	輸入 5 → $1+2+3+4+5=15$ 程式輸出15
10	55	輸入 10 → $1+2+3+4...+10=55$ 程式輸出55



計數迴圈



找到規律簡化



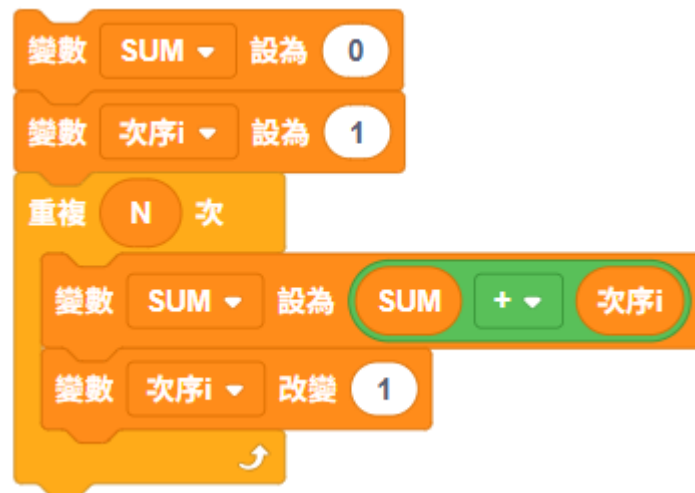
輸入



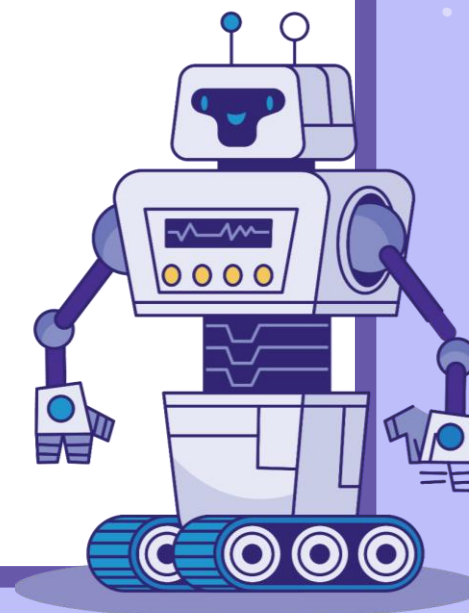
處理



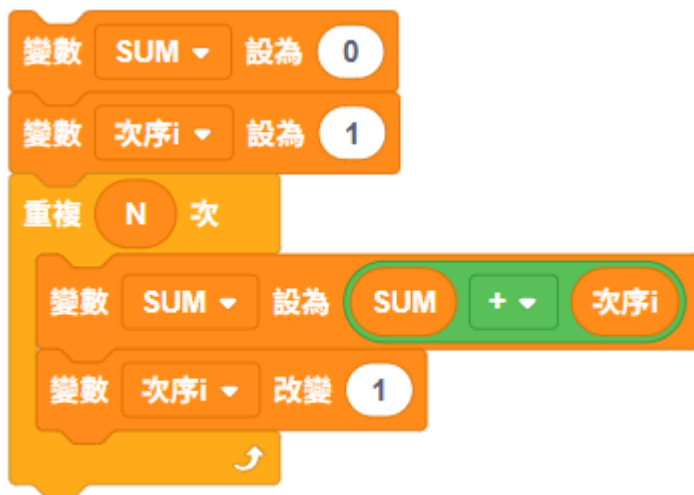
輸出



注意：必須使用迴圈累加



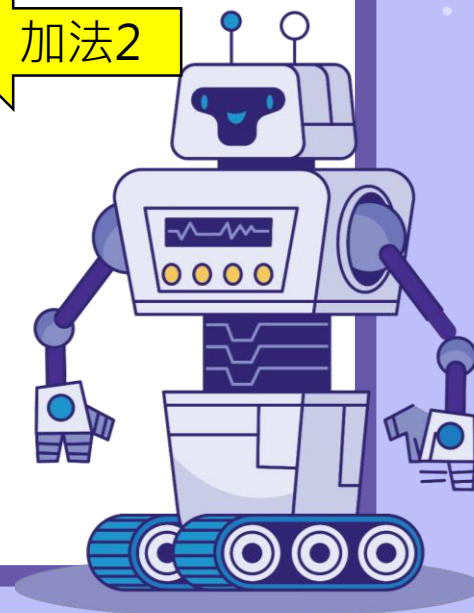
Case Study



← 加法1



← 加法2



輸入

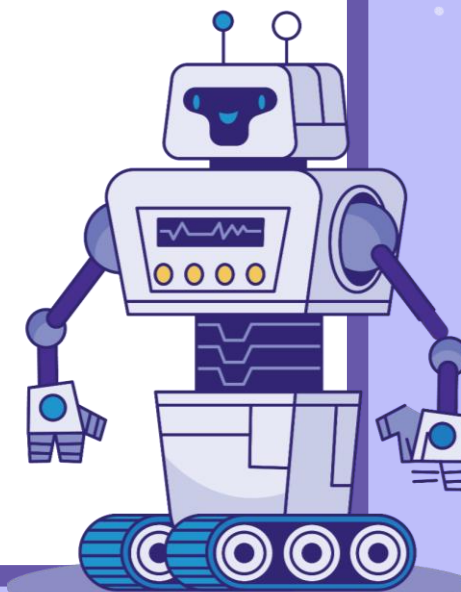


處理

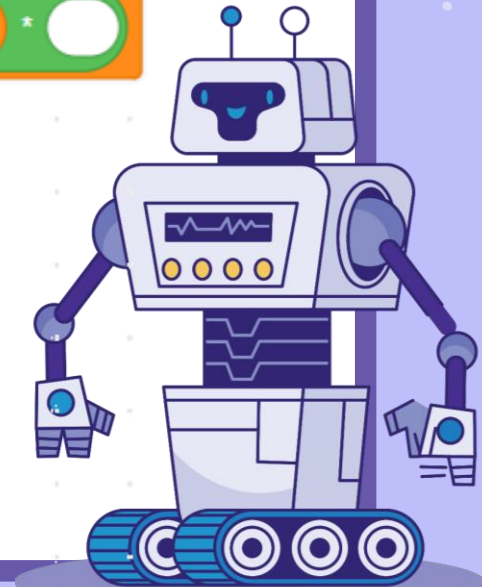
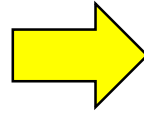


輸出

6-1 換你試試看！



循序結構轉重複結構



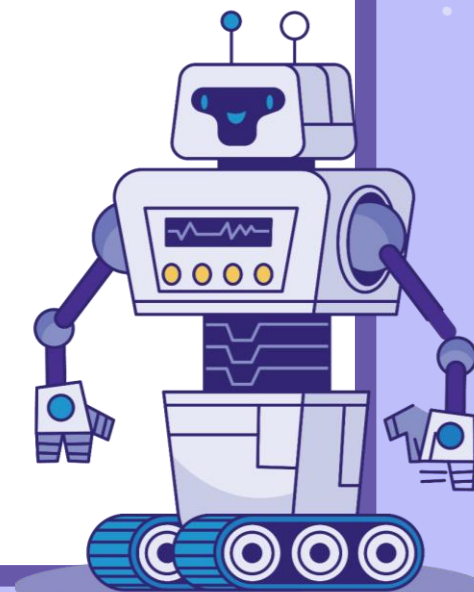
輸入



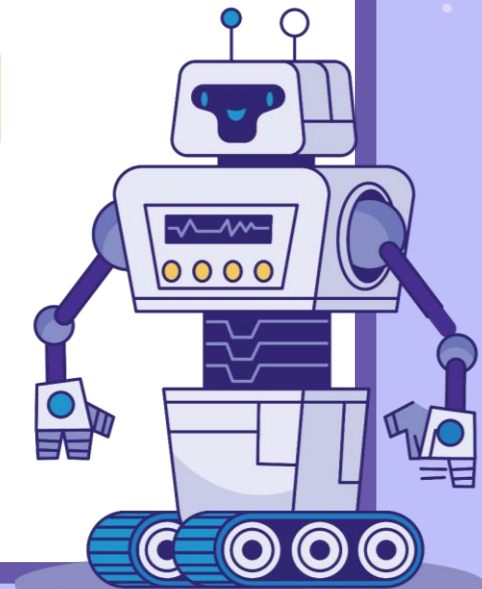
處理



輸出



Case Study-初始值設定



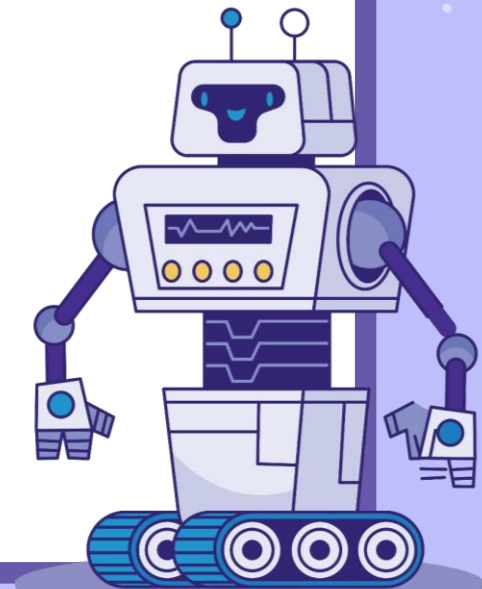
Case Study 有沒有更簡單的方法



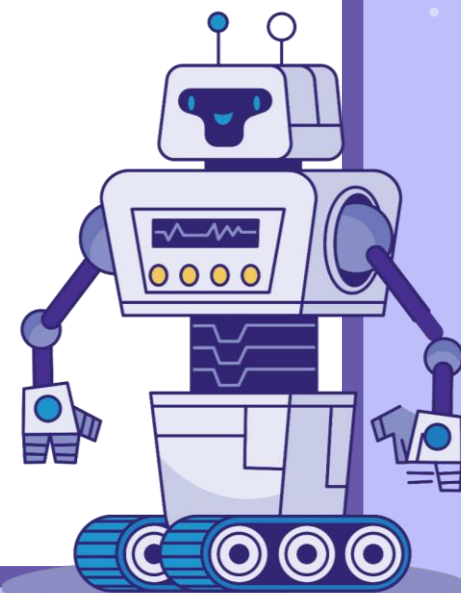
方法1

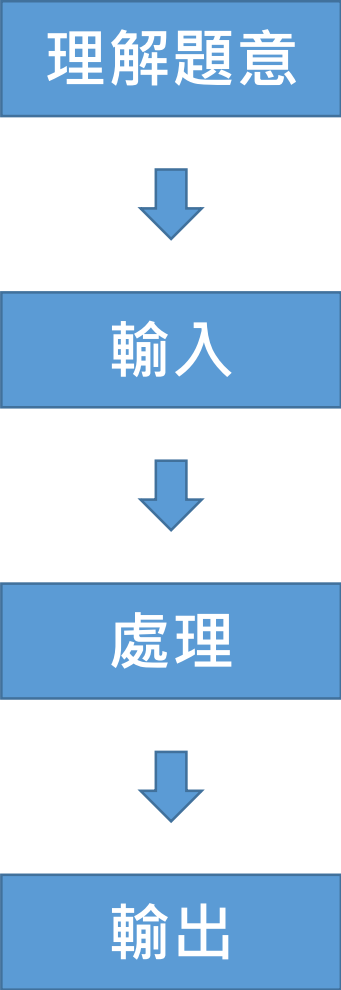


方法2



7-0-乘法表





乘法表

題目說明

請寫一個程式，輸入一個整數 N，輸出 1 到 9 的乘法表結果。

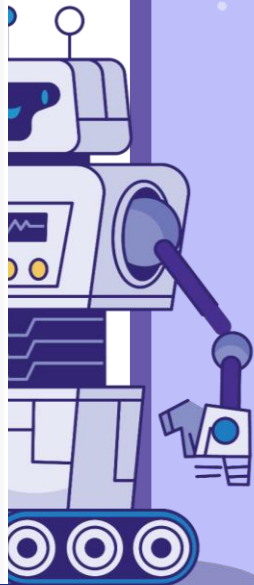
例如輸入 3，輸出 1x3=3, 2x3=6, ..., 9x3=27。

程式輸出3 6 9 12 15 18 21 24 27

這樣的練習能讓你熟悉迴圈的運用。

範例格式

使用者輸入	輸出	範例說明
3	3 6 9 12 15 18 21 24 27	第一行輸入3，經過九九乘法計算後 程式輸出3 6 9 12 15 18 21 24 27
6	6 12 18 24 30 36 42 48 54	第一行輸入6，經過九九乘法計算後 程式輸出6 12 18 24 30 36 42 48 54



文字相加

The image displays a Scratch script and stage illustrating string concatenation. The script on the left consists of eight 'Set Variable to' blocks for a variable named 'SUM'. Each block concatenates a string '字串組合' (String Combination) with the value of 'SUM' multiplied by 3. The strings are '1', '2', '3', '4', '5', '6', and '7' respectively. The stage on the right shows the 'SUM' variable with an orange bar, the Scratch cat sprite, and a robot sprite named 'Sprite1'.

變數 SUM 設為

變數 SUM 設為 字串組合 SUM 字串組合 1 * 3

變數 SUM 設為 字串組合 SUM 字串組合 2 * 3

變數 SUM 設為 字串組合 SUM 字串組合 3 * 3

變數 SUM 設為 字串組合 SUM 字串組合 4 * 3

變數 SUM 設為 字串組合 SUM 字串組合 5 * 3

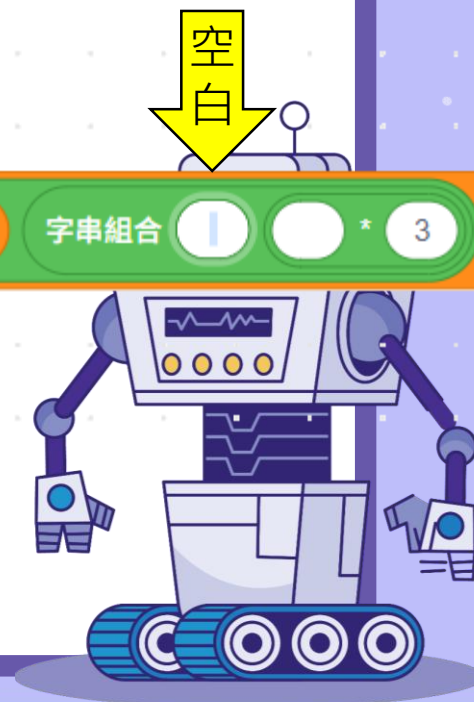
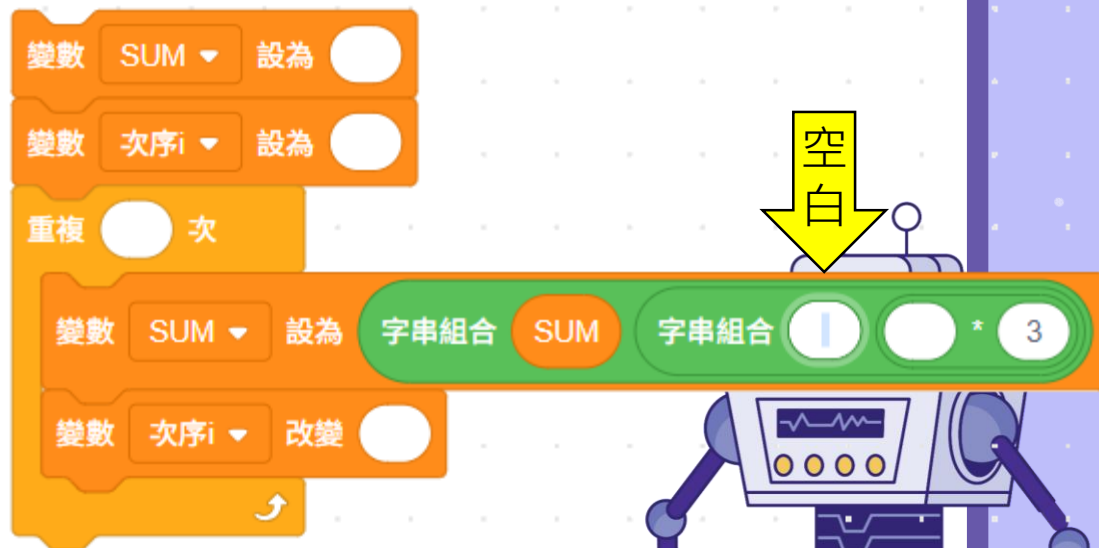
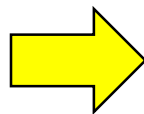
變數 SUM 設為 字串組合 SUM 字串組合 6 * 3

變數 SUM 設為 字串組合 SUM 字串組合 7 * 3

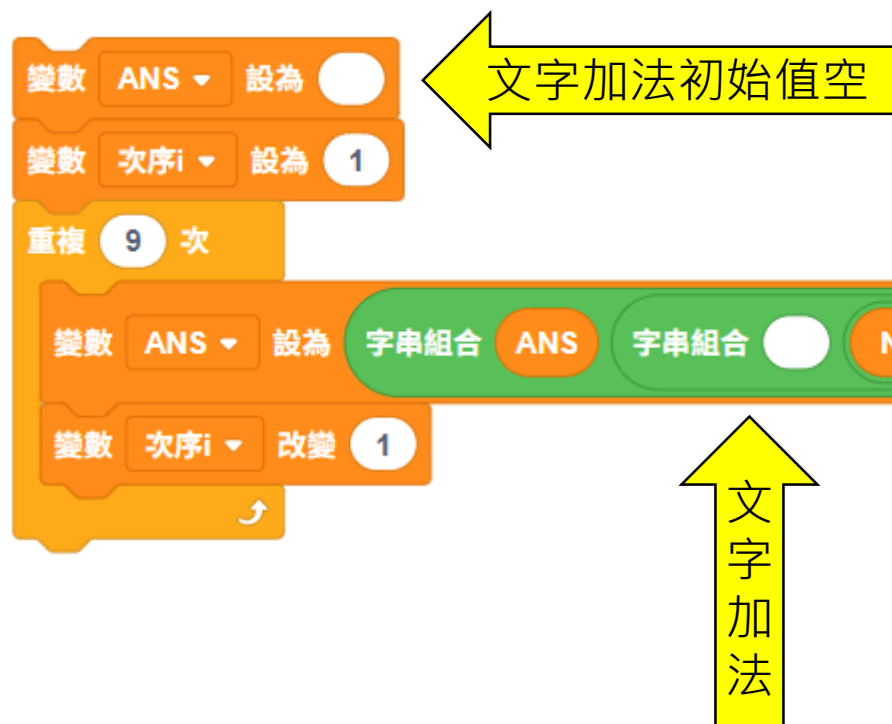
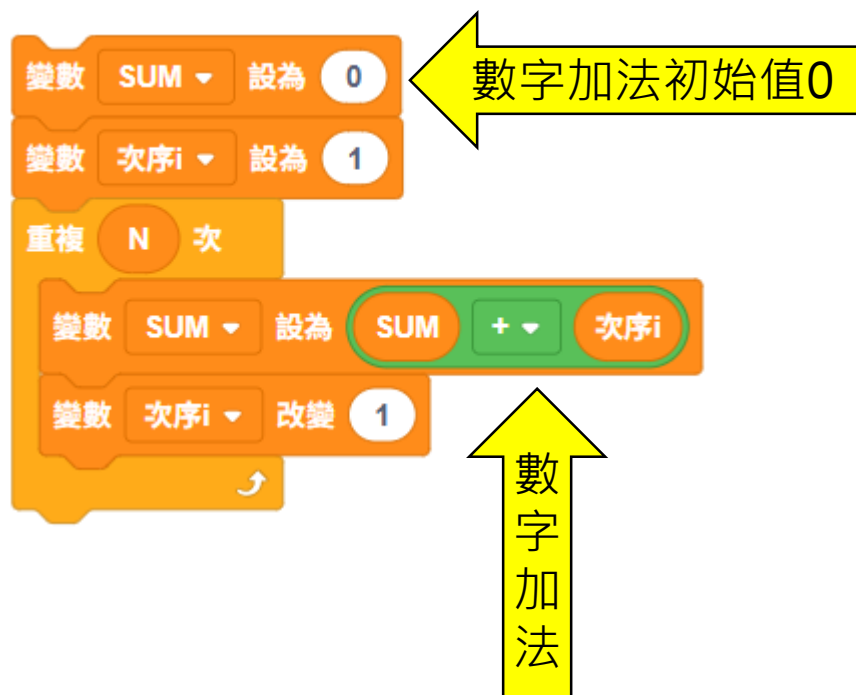
SUM

角色 Sprite1

文字加法&轉成重複結構



Case Study



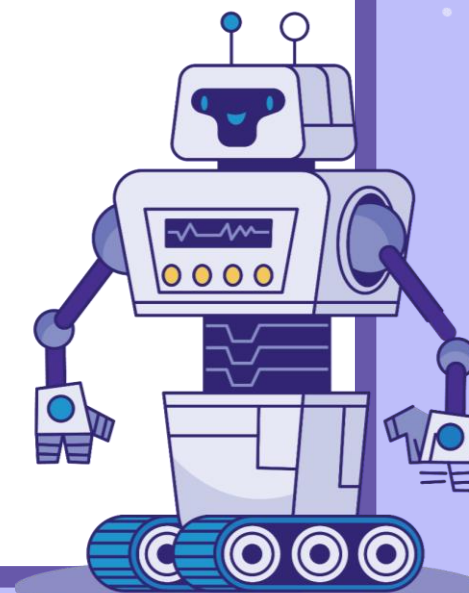
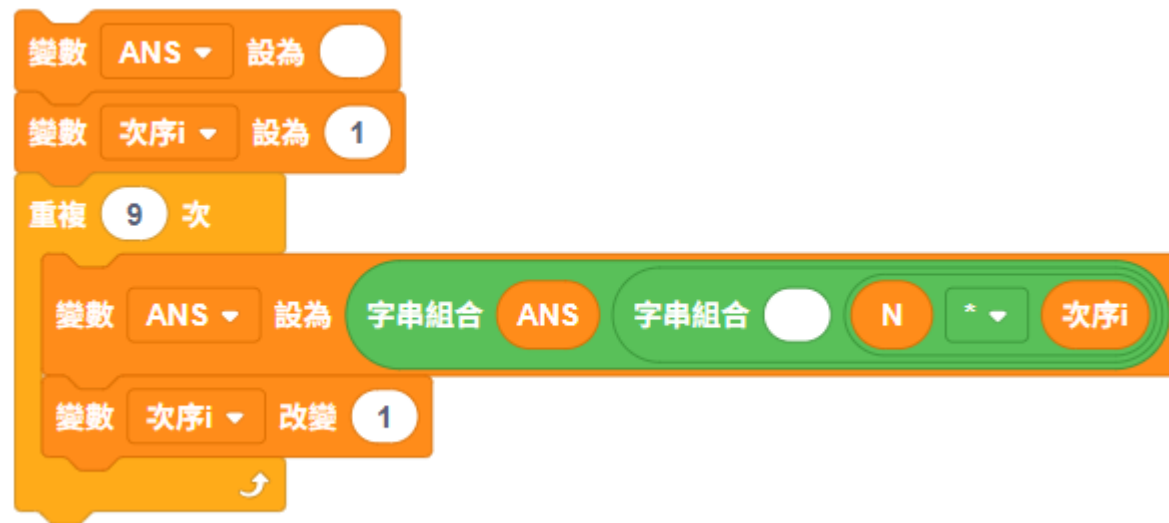
輸入



處理



輸出



輸入

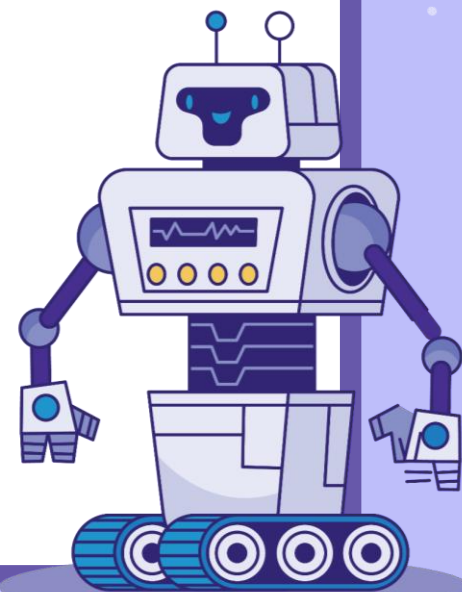


處理



輸出

7-1 換你試試看！



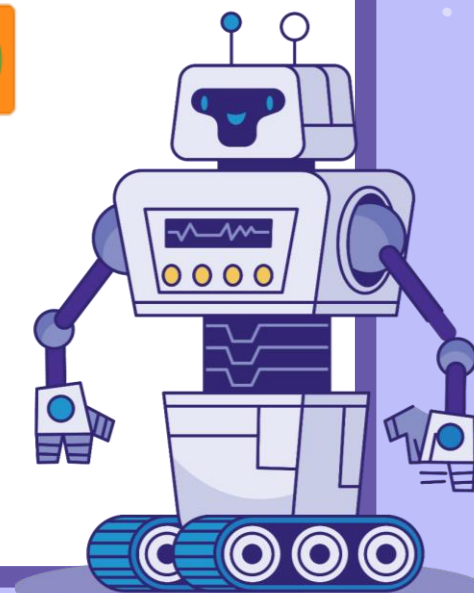
輸入



處理



輸出



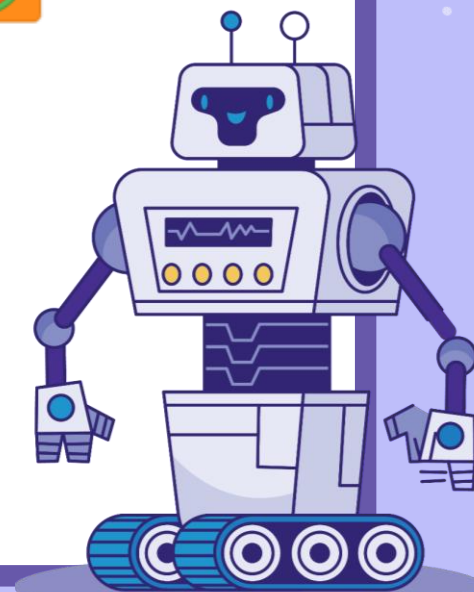
Case Study



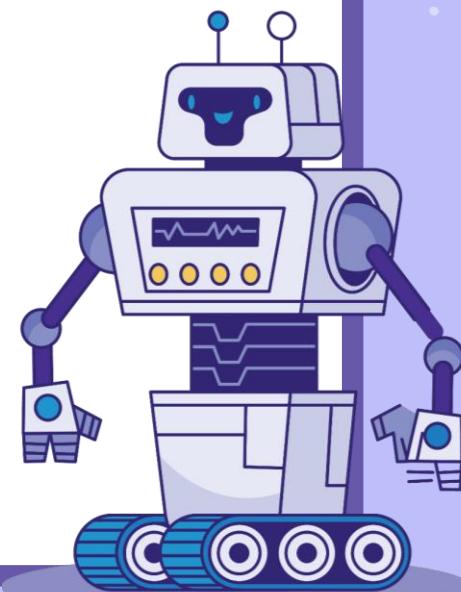
批次輸入



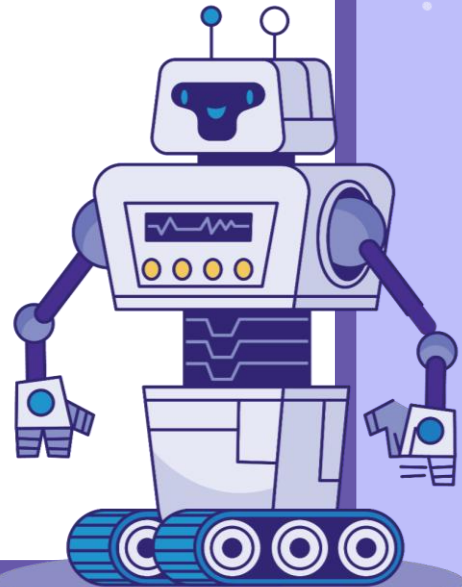
迴圈讀取清單



快充-循環計數迴圈&清單



8-0-找出最大值



理解題意



輸入



處理



輸出

查找最大值

題目說明

小明和朋友一起比賽跑步，老師輸入每位同學的成績，請你寫一個程式找出最高分。

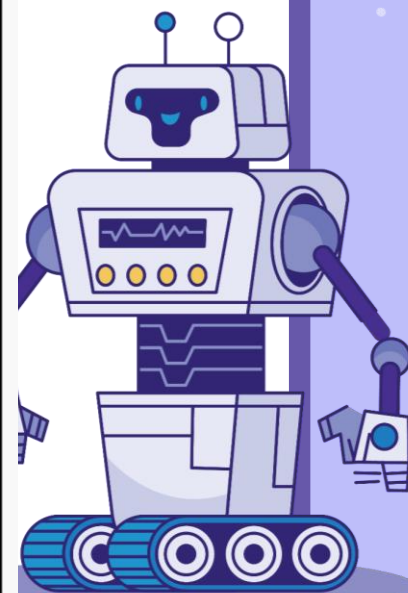
輸入第一行是整數 N ，代表有 N 筆成績

第二行輸入 N 個數字，數字間以空白間隔

程式輸出最大值。這樣的練習可以幫助你處理一串資料，並找出其中的最大值。

範例格式

使用者輸入	輸出	範例說明
3 9 6 8	9	第一行輸入3，表示有3個數字要輸入 第二行輸入9 6 8三個數字序列 程式運算輸出最大值9
6 99 12 129 16 8	129	第一行輸入6，表示有6個數字要輸入 第二行輸入99 12 129 16 8 3六個數字序列 程式運算輸出最大值129



演算法：找出解決問題的
程序及步驟

輸入



處理



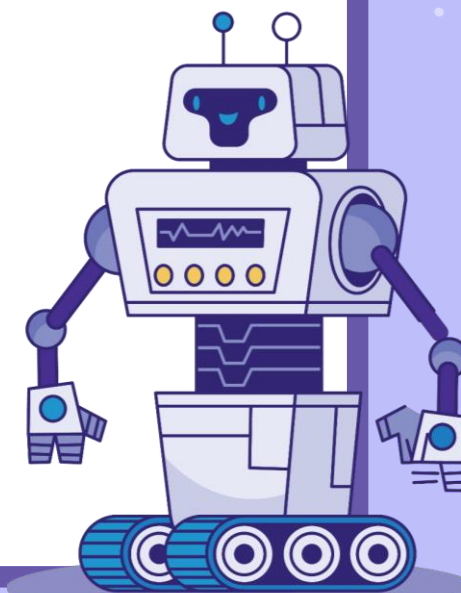
輸出

資料輸入
數字迴圈建立清單

取出清單元素
循環計數

找出最大值

依照題目規定格式輸出
多行輸出
空白間隔輸出



找出最大值演算法

變數 MAX 設為 0

如果 MAX < 樹列清單 的第 1 項 那麼

變數 MAX 設為 樹列清單 的第 1 項

如果 MAX < 樹列清單 的第 2 項 那麼

變數 MAX 設為 樹列清單 的第 2 項

如果 MAX < 樹列清單 的第 3 項 那麼

變數 MAX 設為 樹列清單 的第 3 項

如果 MAX < 樹列清單 的第 4 項 那麼

變數 MAX 設為 樹列清單 的第 4 項

MAX 10

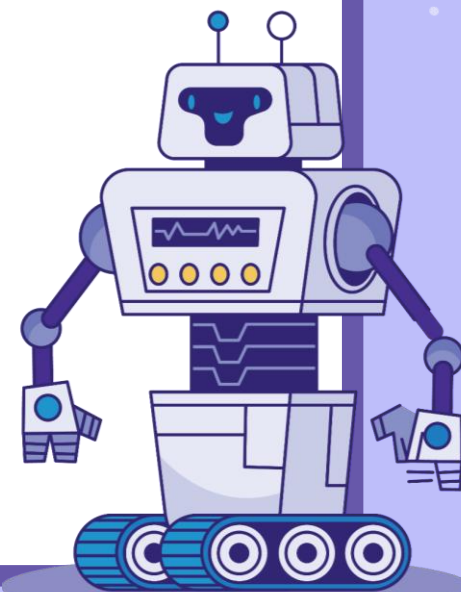
樹列清單

1	3
2	3
3	6
4	3
5	2
6	4
7	9
8	3
9	10
10	8

+ 長度 10 =

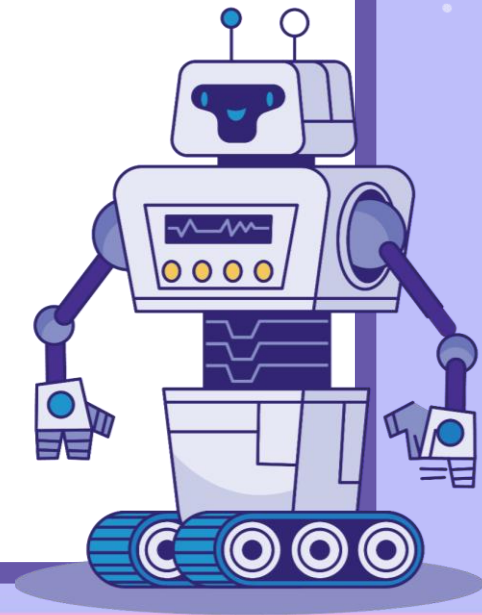
角色 Sprite1 x 109 y 10

顯示 尺寸 100 方向 90

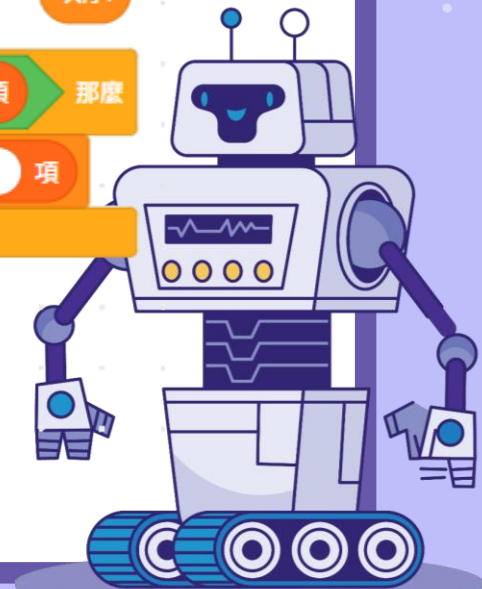
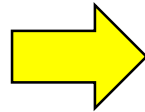


找最大值的演算法

- 先設定最大值MAX的初始值(0 OR待比較的清單第一項)
- 用迴圈依序叫出清單每一項跟MAX比較
- 如果MAX小於清單第i項，MAX設為清單第i項
- 跑完迴圈，MAX就是該清單元素的最大值



循序結構轉重複結構



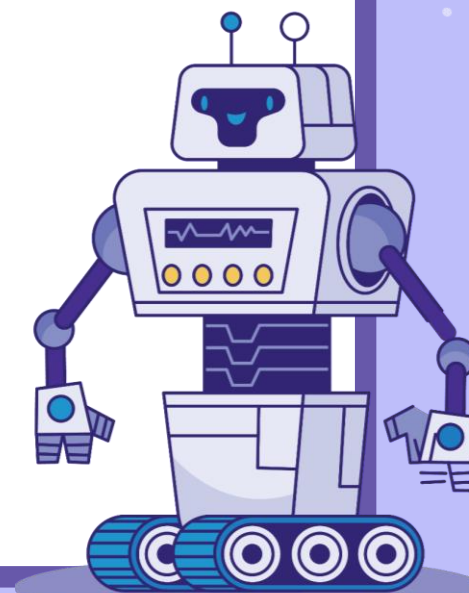
輸入



處理



輸出



輸入

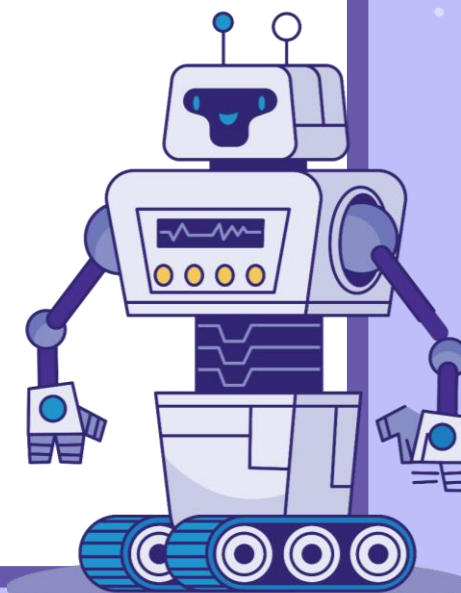


處理



輸出

8-1 換你試試看！



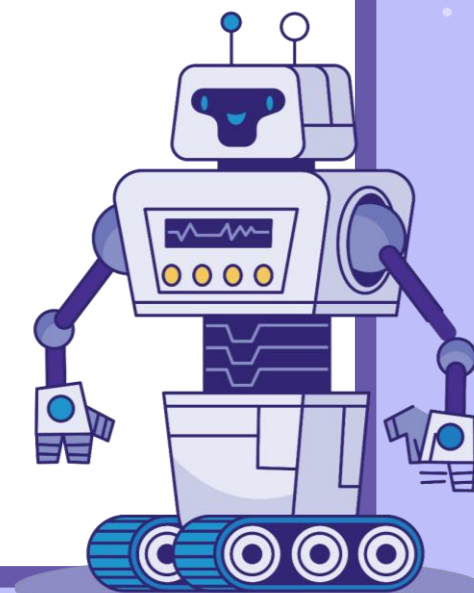
輸入



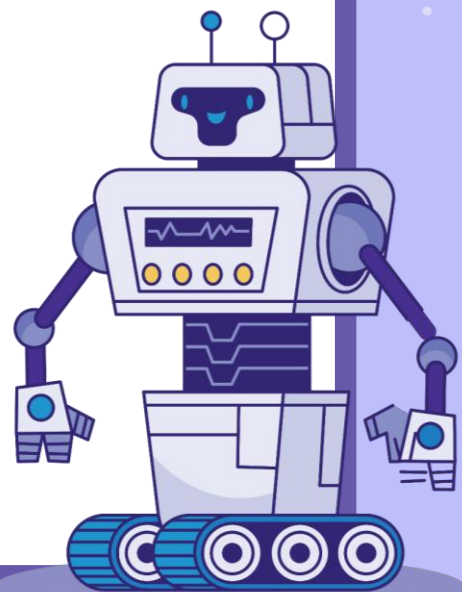
處理



輸出



9-0-判斷質數



理解題意



輸入



處理



輸出

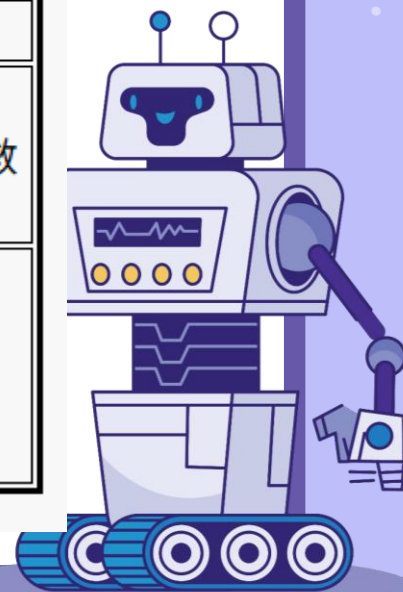
找質數

題目說明

質數是大於 1 且只能被 1 和自己整除的數字。
請寫一個程式，輸入一個整數 N，判斷它是否為質數。
若是質數輸出 Yes，否則輸出 No。
這題訓練你使用條件與迴圈判斷。

範例格式

使用者輸入	輸出	範例說明
7	Yes	第一行輸入7 電腦運算判斷7是質數 程式運算輸出Yes
6	No	第一行輸入6 電腦運算判斷6可被2 整除，不是質數 程式運算輸出No



輸入



處理



輸出

想想看，如何判斷是不是質數？

01

1

檢查數字大小

確認數字是否大於1。如果小於或等於1，則不是質數。2是最小的質數，也是唯一的偶數質數。

02

2

排除偶數

除了2以外，所有偶數都不是質數，因為它們可以被2整除。這個步驟可以快速排除一半的候選數字。

03

試除法檢測

3

用小於該數平方根的所有質數去除它。如果都無法整除，則該數為質數。例如：檢查29時，只需試除到5即可。

04

確認結果

若以上測試都通過，該數即為質數。記住：質數在數學和密碼學中都扮演重要角色！

取開根號無條件進位



why開根號？

- $a * a = b \quad a = \sqrt{b}$

1 2 3 6 7 14 21 42

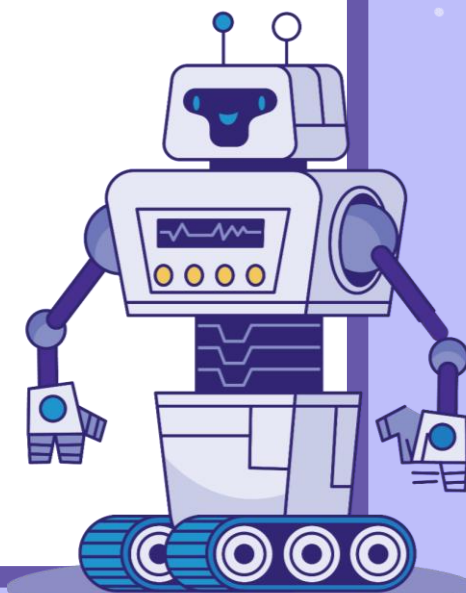
$$\sqrt{42} = 6.x$$

無條件進位 ▾

開根號 ▾

數值

42



輸入



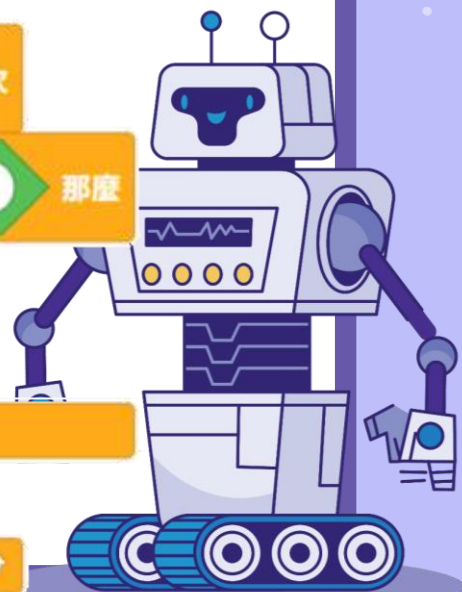
處理



輸出



從2開始，Why？



輸入

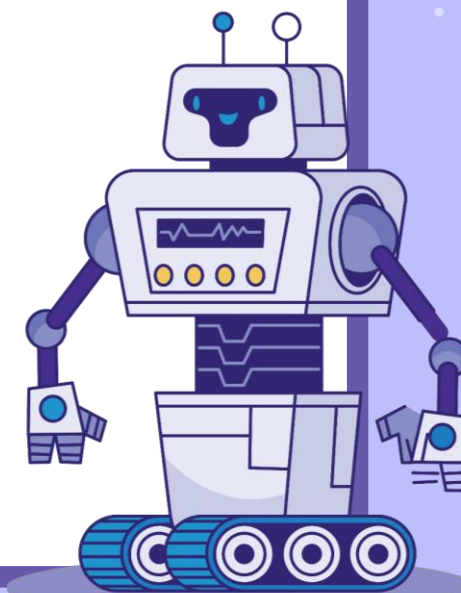


處理



輸出

9-1 換你試試看！



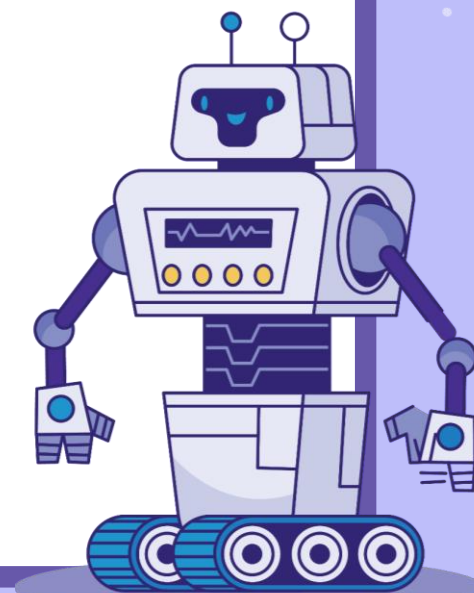
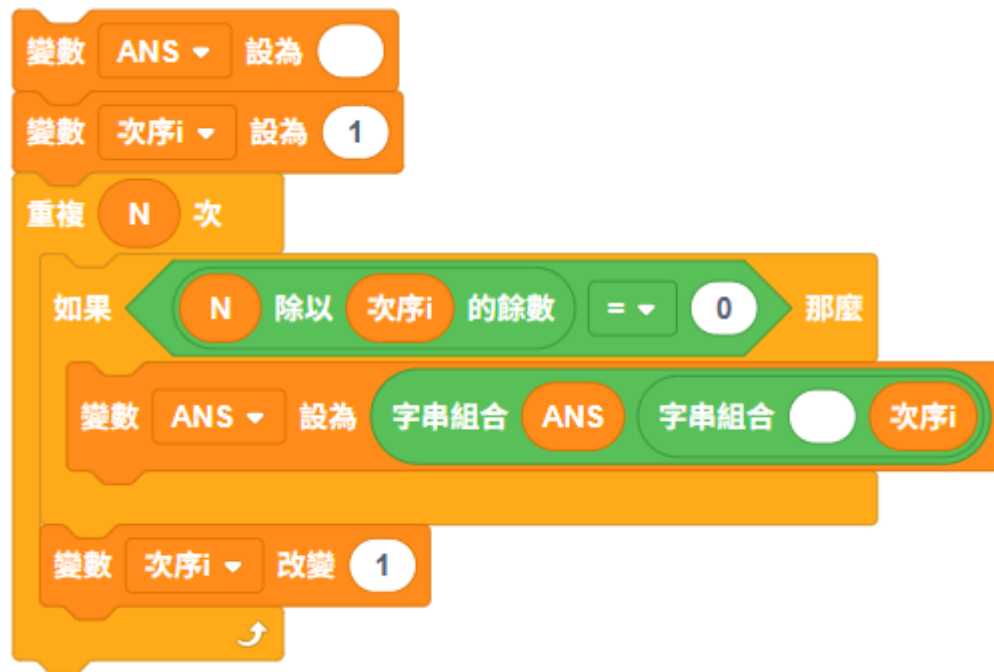
輸入



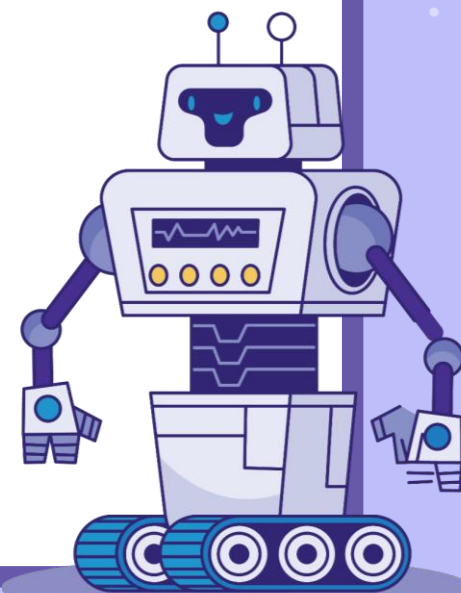
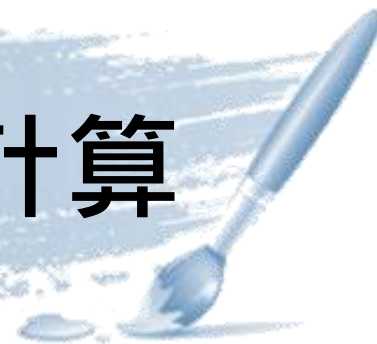
處理



輸出



10-0-多科成績計算



多科成績計算

題目說明

小明參加了數學、英文、自然...等多科的段考，老師希望能快速算出總分與平均分數，並判斷是否及格。

請設計程式，輸入多科的成績，計算總分與平均。

第一行輸入共有幾科

第二行依序輸入各科成績，科成績之間以空白間隔

計算總分，平均成績以四捨五入取整數

如果平均分數大於等於 60，就輸出「及格」，否則輸出「不及格」。

這題可以訓練你如何處理多筆資料的加總、平均計算，並搭配條件判斷。

範例格式

使用者輸入	輸出	範例說明
3 80 70 90	240 80 及格	第一行輸入80 70 90，經過計算後總分180、平均80、及格
6 50 41 60 55 70 65	341 57 不及格	第一行輸入50 41 60 55 70 65，經過計算後 程式輸出總分341、平均57、不及格

理解題意



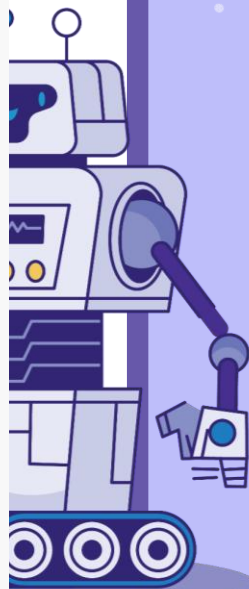
輸入



處理



輸出



演算法：找出解決問題的
程序及步驟

輸入



處理



輸出

資料輸入

文本建立清單
數字迴圈建立清單

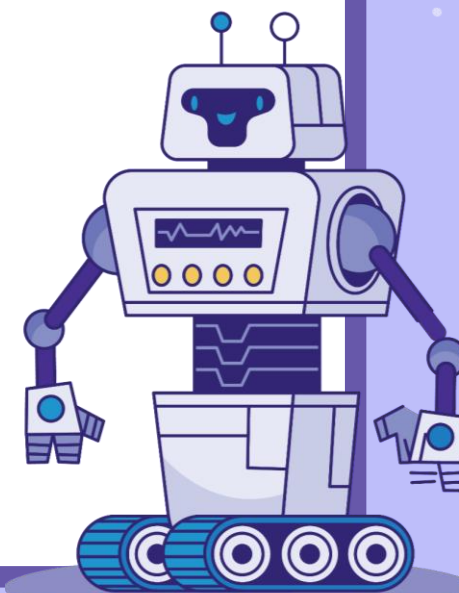
取出清單元素

循環計數

總分、平均、判斷及格

依照題目規定格式輸出

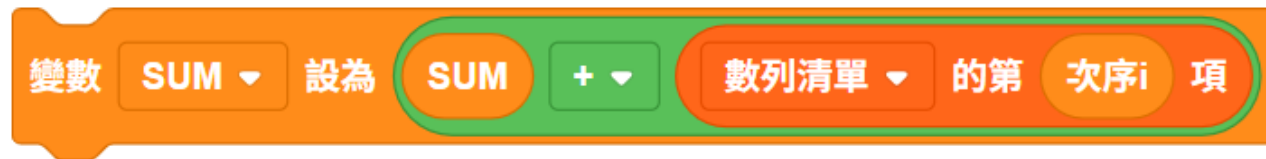
多行輸出
空白間隔輸出



技能整合



批次輸入



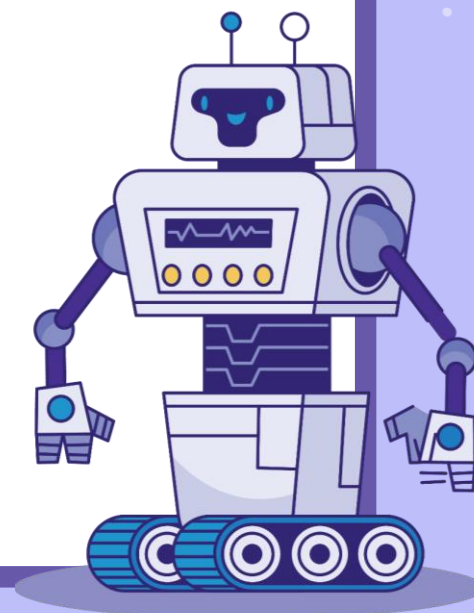
數字加法



計數迴圈



選擇結構



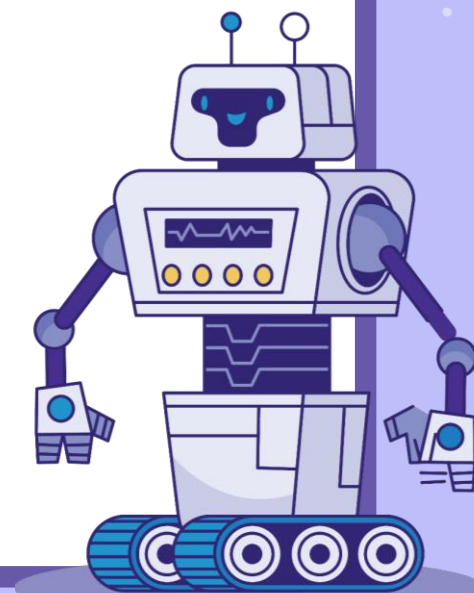
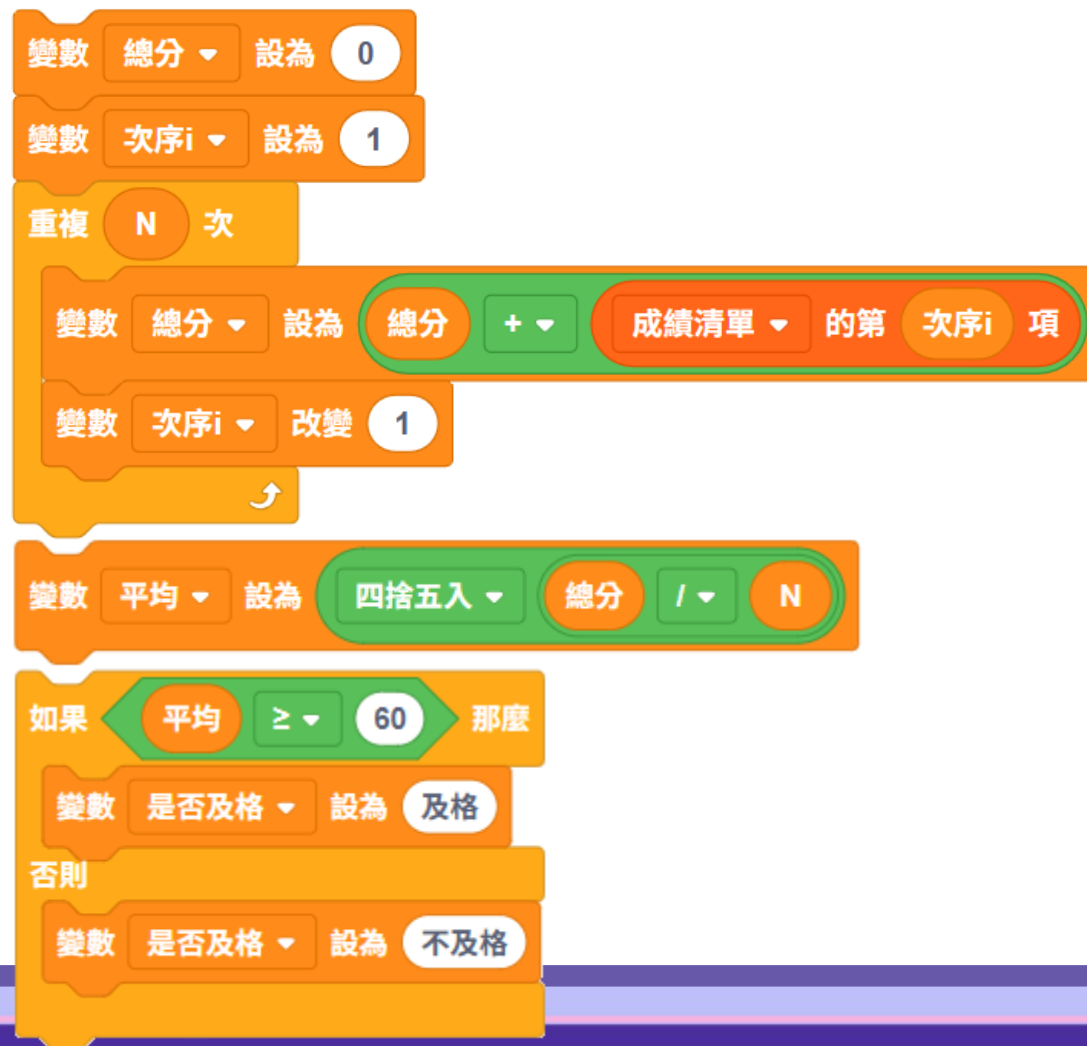
輸入



處理



輸出



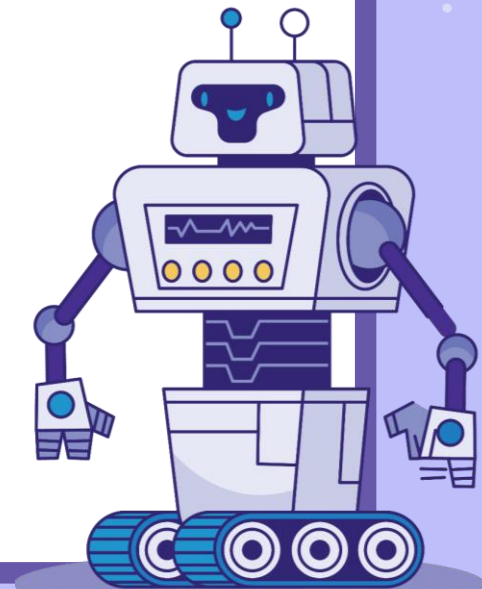
Case Study-哪一種比較好呢？



循環計數迴圈



重複迴圈



輸入

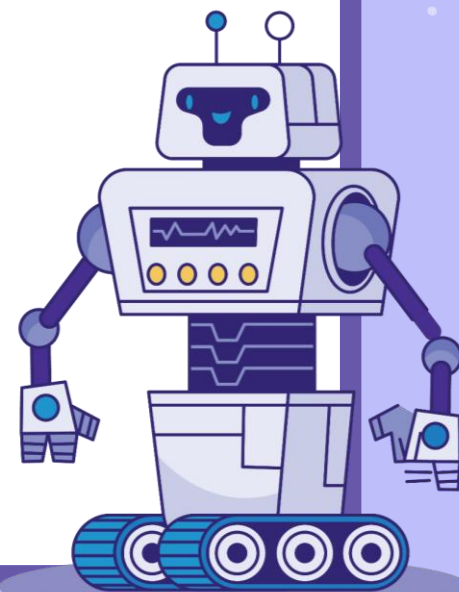


處理



輸出

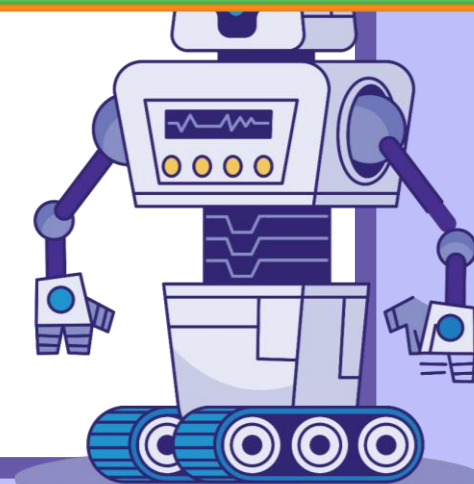
10-1 換你試試看！



輸入



處理1



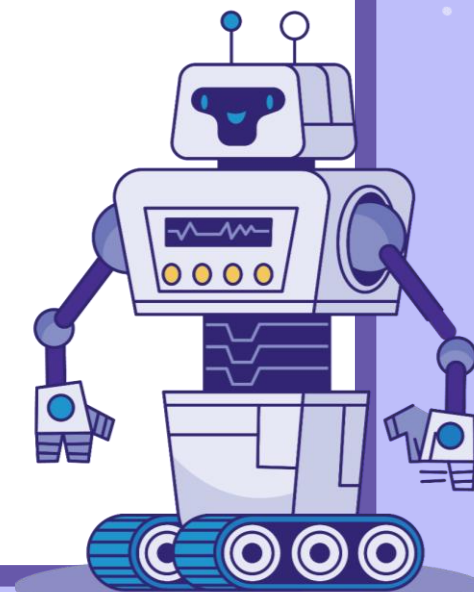
處理1



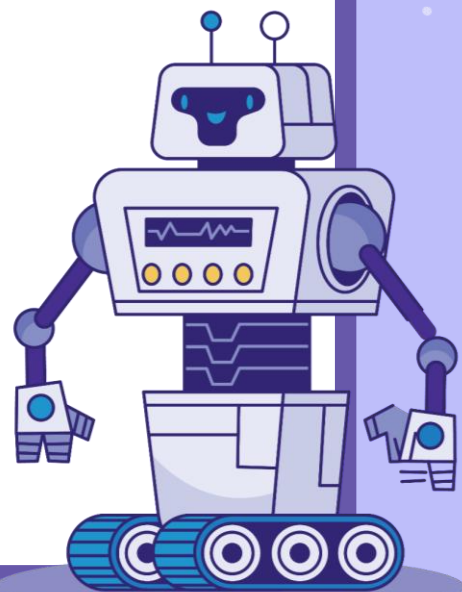
處理2



輸出



11-0-計算字元出現的次數



理解題意



輸入



處理



輸出

計算字元出現次數

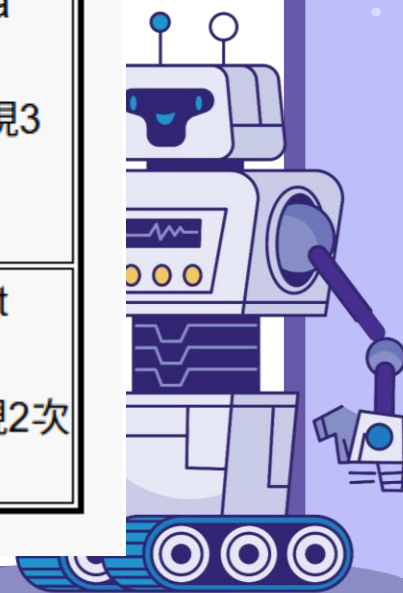
題目說明

請設計程式，輸入一個字串，並輸入一個英文字母輸出該字母在字串中出現的次數。

這樣的練習可以訓練你操作字串與統計。

範例格式

使用者輸入	輸出	範例說明
banana a	3	第一行輸入banana 第二行輸入字元a 程式運算比對a出現3次 程式輸出3
student t	2	第一行輸入student 第二行輸入字元t 程式運算比對t出現2次 程式輸出2



演算法：找出解決問題的 程序及步驟

輸入



處理



輸出

資料輸入

輸入待比對字串
輸入要比對的字元

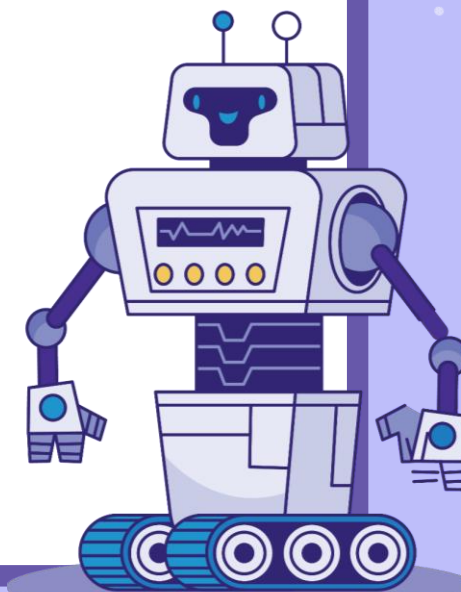
取出字串每一個元素

循環計數

比對統計字元出現次數

依照題目規定格式輸出

多行輸出
空白間隔輸出



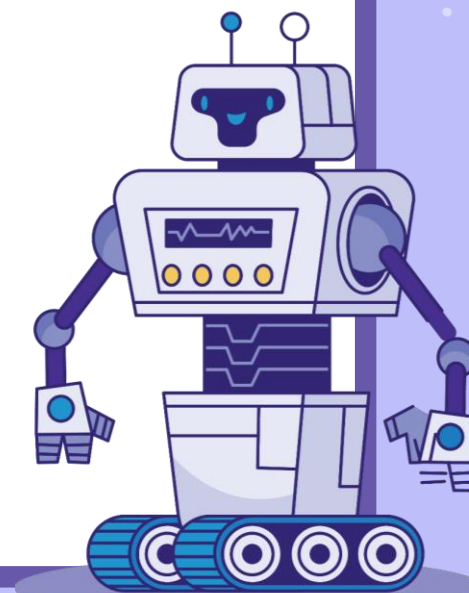
輸入



處理



輸出



輸入

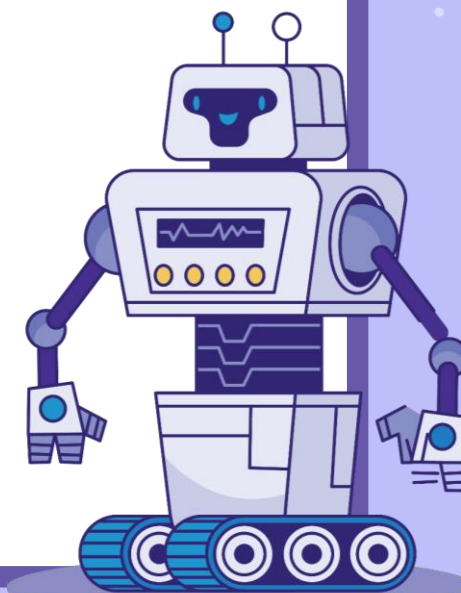


處理



輸出

11-1 換你試試看！



演算法：找出解決問題的
程序及步驟

輸入



處理



輸出

資料輸入

輸入待比對字串
輸入要比對的字元？

依序取出字串要比對的元素

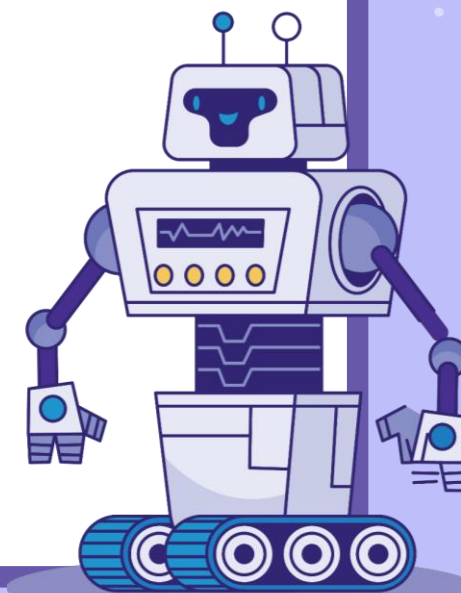
循環計數

逐一取出字串待比對的元素

逐一比對統計字元出現次數(

依照題目規定格式輸出

多行輸出
空白間隔輸出



輸入

處理

輸出

當 被點擊

詢問 並等待

變數 輸入字串 設為 詢問的答案

1 變數 最多次數MAX 設為 0

變數 次序i 設為 1 3

重複 取得 輸入字串 的長度 次

統計字元出現次數 字串 輸入字串 的第 次序i 字

1 如果 最多次數MAX < 字元出現次數 那麼

變數 最多次數MAX 設為 字元出現次數

變數 次序i 改變 1

2 定義 統計字元出現次數 待比對字元

變數 字元出現次數 設為 0

變數 次序j 設為 1 3

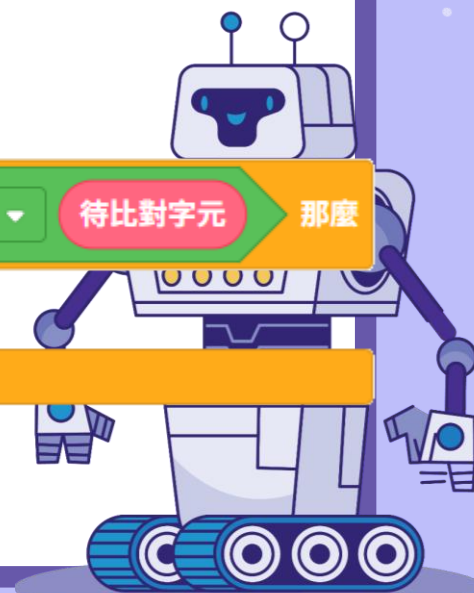
重複 取得 輸入字串 的長度 次

如果 字串 輸入字串 的第 次序j 字 = 待比對字元 那麼

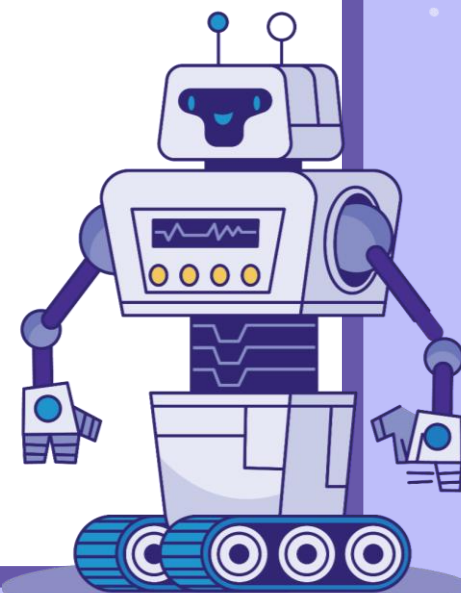
變數 字元出現次數 改變 1

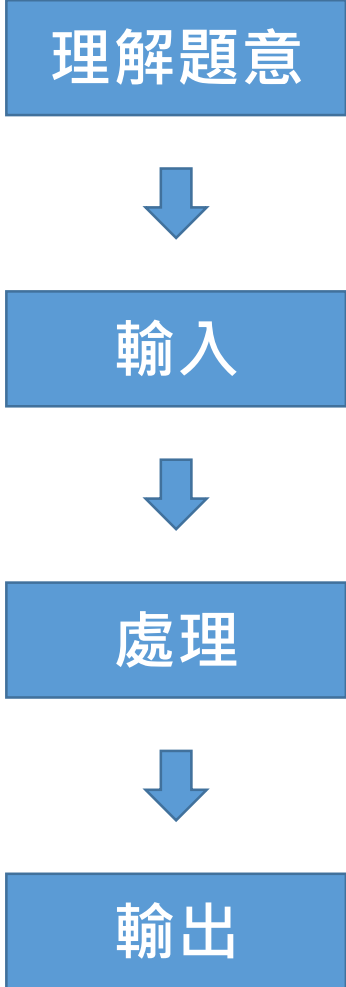
變數 次序j 改變 1

說出 最多次數MAX



12-0-簡易密碼轉換





簡易密碼轉換

題目說明

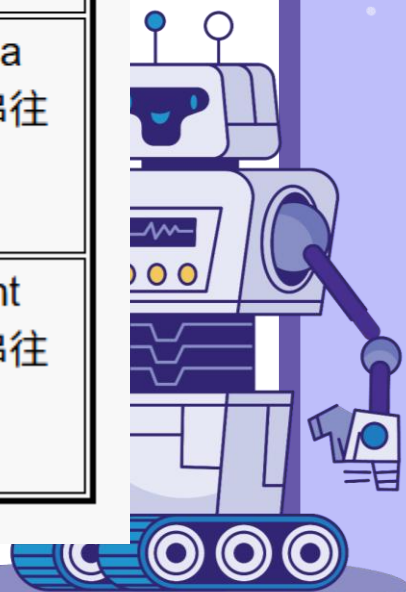
小明設計了一種簡單的密碼規則：將每個字母向後移二個字母（z 變 a）。

請寫一個程式，輸入一個英文單字，輸出轉換後的字串。

這樣的練習訓練你字元處理與條件轉換。

範例格式

使用者輸入	輸出	範例說明
banana	dcpcpc	第一行輸入banana 程式運算，將字串往後移2個位置加密 程式輸出dcpcpc
student	uvwfgpv	第一行輸入student 程式運算，將字串往後移2個位置加密 程式輸出uvwfgpv



演算法：找出解決問題的
程序及步驟

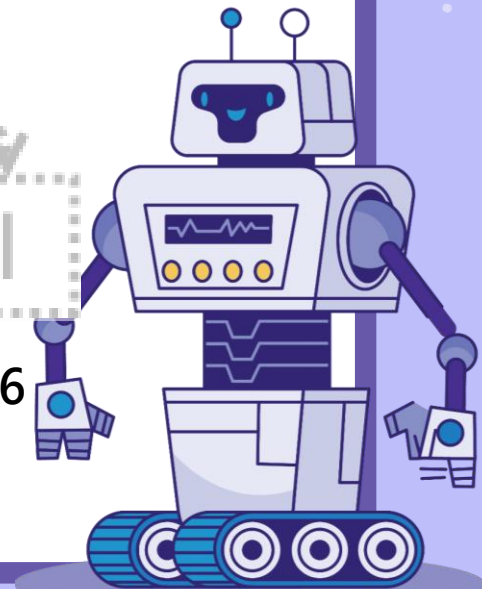
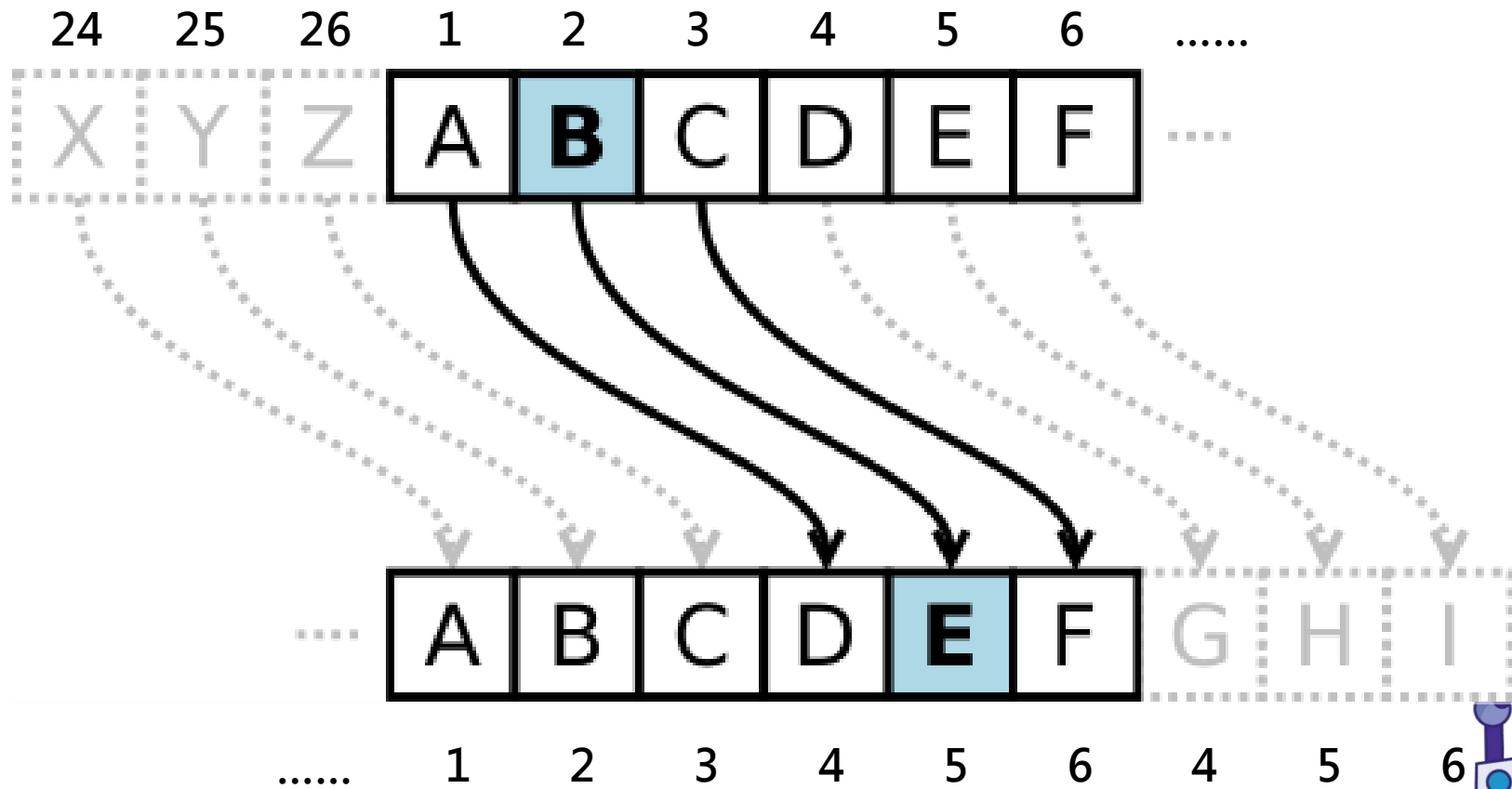
輸入



處理



輸出



演算法：找出解決問題的 程序及步驟

輸入



處理



輸出

資料輸入

輸入原始密碼字串

依序取出字元比對密碼環

循環計數

① 找出字元在清單的編號

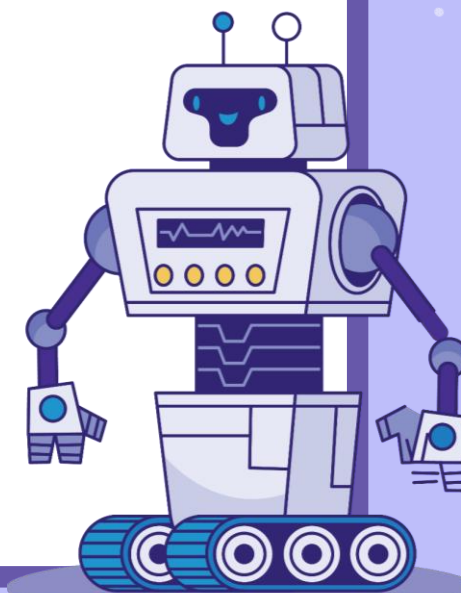
取得轉換後(+2)的字元

② 編號+2取得清單字元

③ 超過26的處理

依照題目規定格式輸出

④ 多行輸出
空白間隔輸出



輸入



處理



輸出

當 被點擊

詢問 並等待

變數 輸入字串 設為 詢問的答案

1-建立密碼環清單

定義 1-建立密碼環清單

變數 密碼環 設為 abcdefghijklmnopqrstuvwxyz

變數 次序i 設為 1

重複 26 次

添加 字串 密碼環 的第 次序i 字 到 密碼環清單

變數 次序i 改變 1

變數 ANS 設為

變數 次序j 設為 1

重複 取得 輸入字串 的長度 次

1 變數 原始次序 設為 字串 輸入字串 的第 次序j 字 在 密碼環清單 裡的項目編號

2 變數 轉換後次序 設為 原始次序 + 2

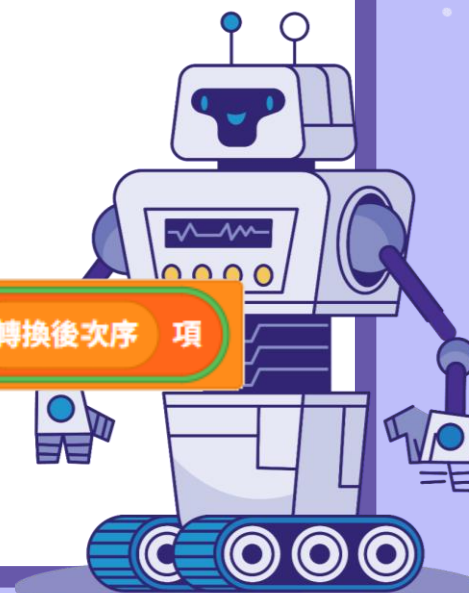
3 如果 轉換後次序 > 26 那麼

變數 轉換後次序 改變 -26

4 變數 ANS 設為 字串組合 ANS 密碼環清單 的第 轉換後次序 項

變數 次序j 改變 1

說出 ANS



輸入

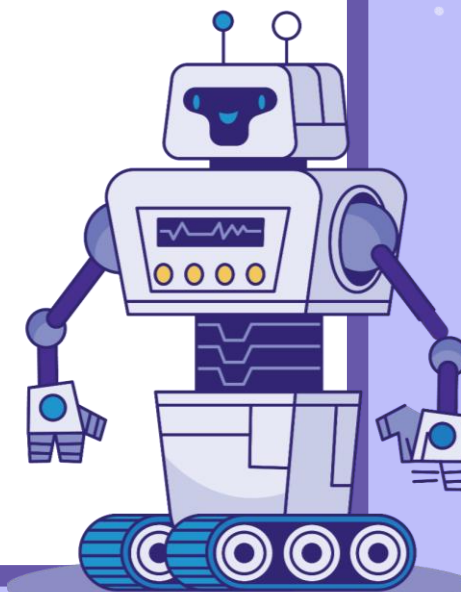


處理



輸出

12-1 換你試試看！



輸入



處理



輸出

當 旗幟 被點擊

詢問 並等待

變數 N 設為 詢問的答案

詢問 並等待

變數 輸入字串 設為 詢問的答案

1-建立密碼環清單

定義 1-建立密碼環清單

變數 密碼環 設為 abcdefghijklmnopqrstuvwxyz0123456789

變數 次序i 設為 1

重複 36 次

添加 字串 密碼環 的第 次序i 字 到 密碼環清單

變數 次序i 改變 1

變數 ANS 設為

變數 次序j 設為 1

重複 取得 輸入字串 的長度 次

1 變數 原始次序 設為 字串 輸入字串 的第 次序j 字 在 密碼環清單 裡的項目編號

2 變數 轉換後次序 設為 原始次序 - N

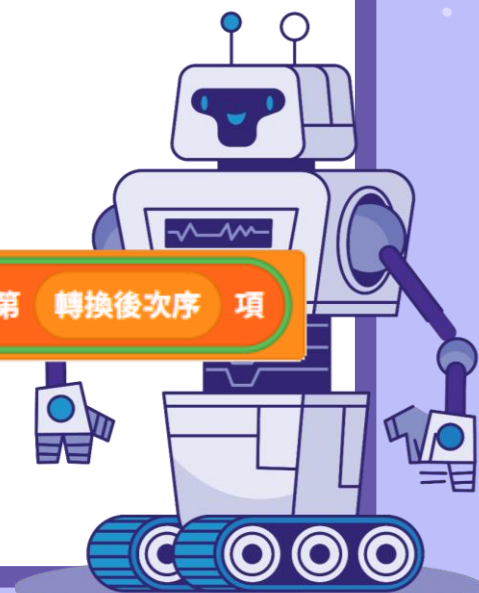
3 如果 轉換後次序 < 1 那麼

變數 轉換後次序 改變 36

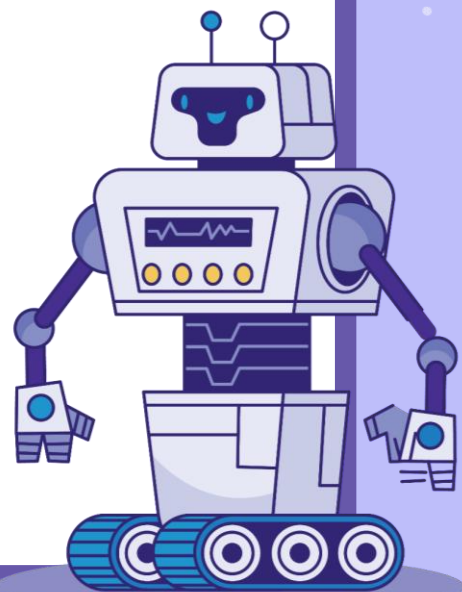
4 變數 ANS 設為 字串組合 ANS 密碼環清單 的第 轉換後次序 項

變數 次序j 改變 1

說出 ANS



13-0-二數的最大公因數



理解題意



輸入



處理



輸出

二數的最大公因數

題目說明

小明上學時，不太會找最大公因數，請寫個程式幫他找一下

輸入二個整數，請計算這二個數字的最大公因數

輸入格式：輸入2行

第一行輸入第1個整數N

第二行輸入地2個整數M

程式運算找出二數的最大公因數並輸出

範例格式

使用者輸入	輸出	範例說明
9 6	3	第一行輸入9， 第二行輸入6， 程式輸出6、9二個數 的最大公因數3
30 72	6	第一行輸入30， 第二行輸入72 程式輸出30，72二個 數的最大公因數6

演算法：找出解決問題的
程序及步驟

輸入



處理



輸出

暴力解法

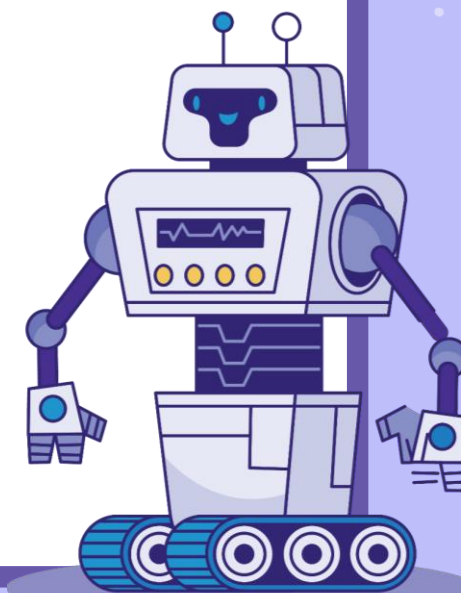
資料輸入

依序比對是否整除

循環計數
由大至小，找到即中斷

依照題目規定格式輸出

多行輸出
空白間隔輸出



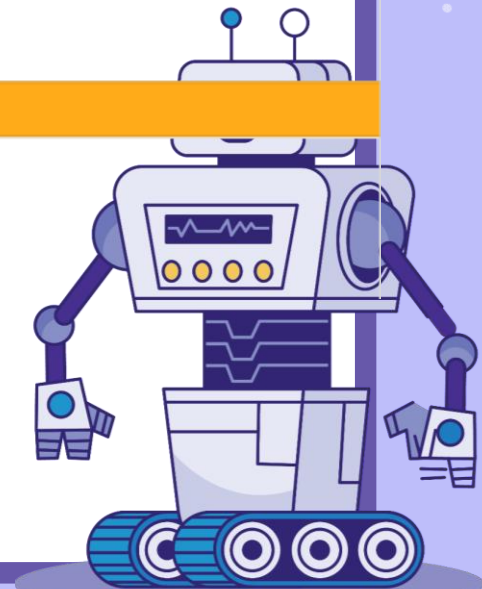
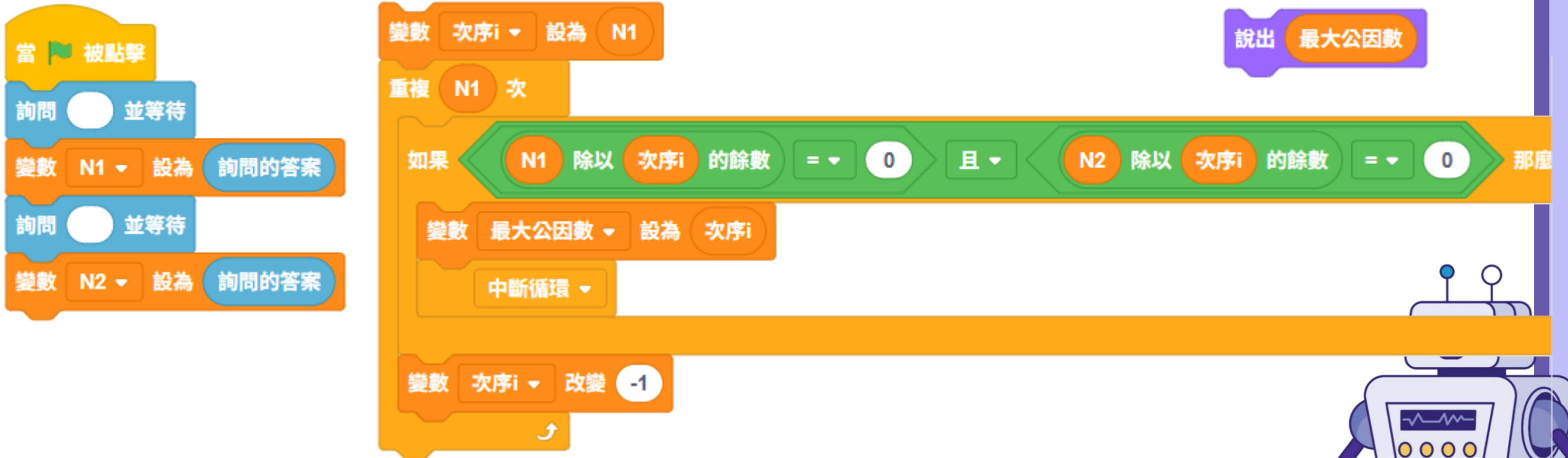
輸入



處理

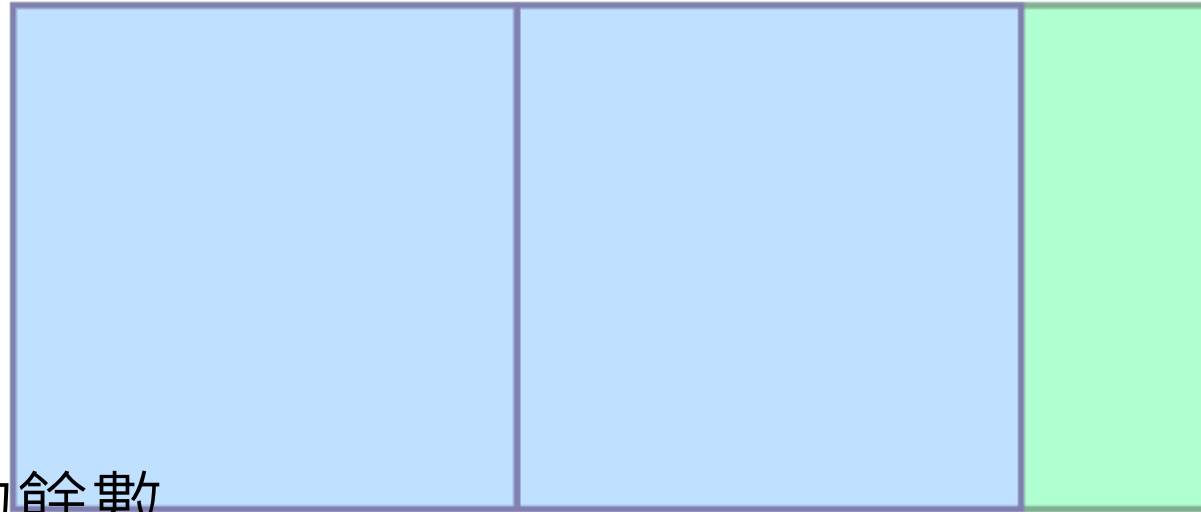


輸出



輾轉相除法

- 除數 b ，被除數 a
- 重複到 b 整除 a
- 餘數 = 被除數 / 除數的餘數
- 被除數 = 除數
- 除數 = 餘數
- 除數即為最大公因數



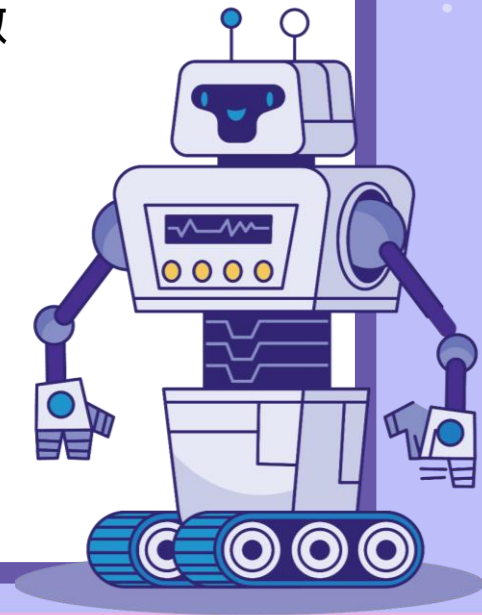
a 被除數

b 除數

r 餘數



嘗試以 16×16 的正方形鋪滿。



輸入



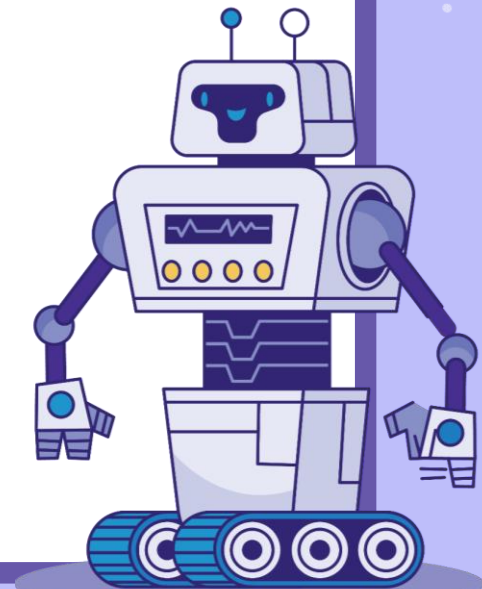
處理



輸出



說出 最大公因數



輸入

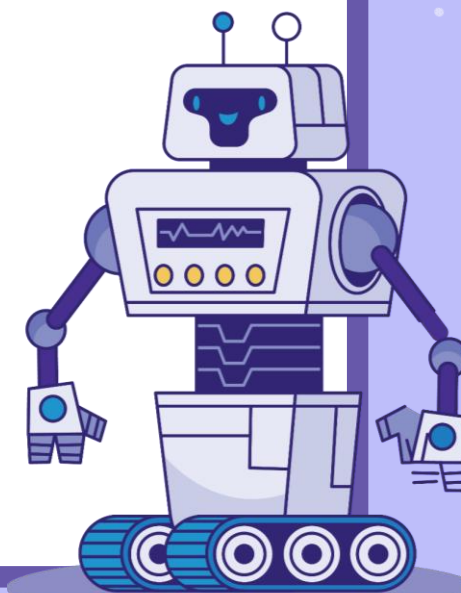


處理



輸出

13-1 換你試試看！



輸入



處理



輸出

當 被點擊

詢問 並等待

變數 N 設為 詢問的答案

重複 N 次

詢問 並等待

添加 詢問的答案 到 數列清單

變數 最大公因數 設為 1

變數 次序i 設為 數列清單 的第 1 項

重複 數列清單 的第 1 項 次

逐一比對

如果 TEMP = N 那麼

變數 最大公因數 設為 次序i

中斷循環

變數 次序i 改變 -1

定義 逐一比對

變數 TEMP 設為 0

變數 次序j 設為 1

重複 N 次

如果 數列清單 的第 次序j 項 除以 次序i 的餘數 $\neq 0$ 那麼

中斷循環

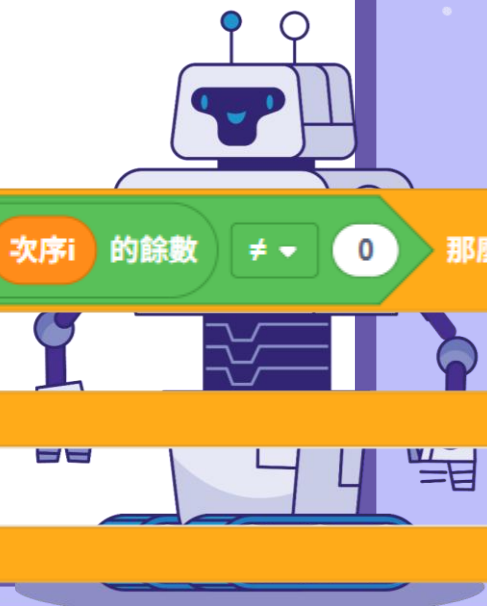
否則

變數 TEMP 改變 1

變數 次序j 改變 1

說出 最大公因數

暴力解法



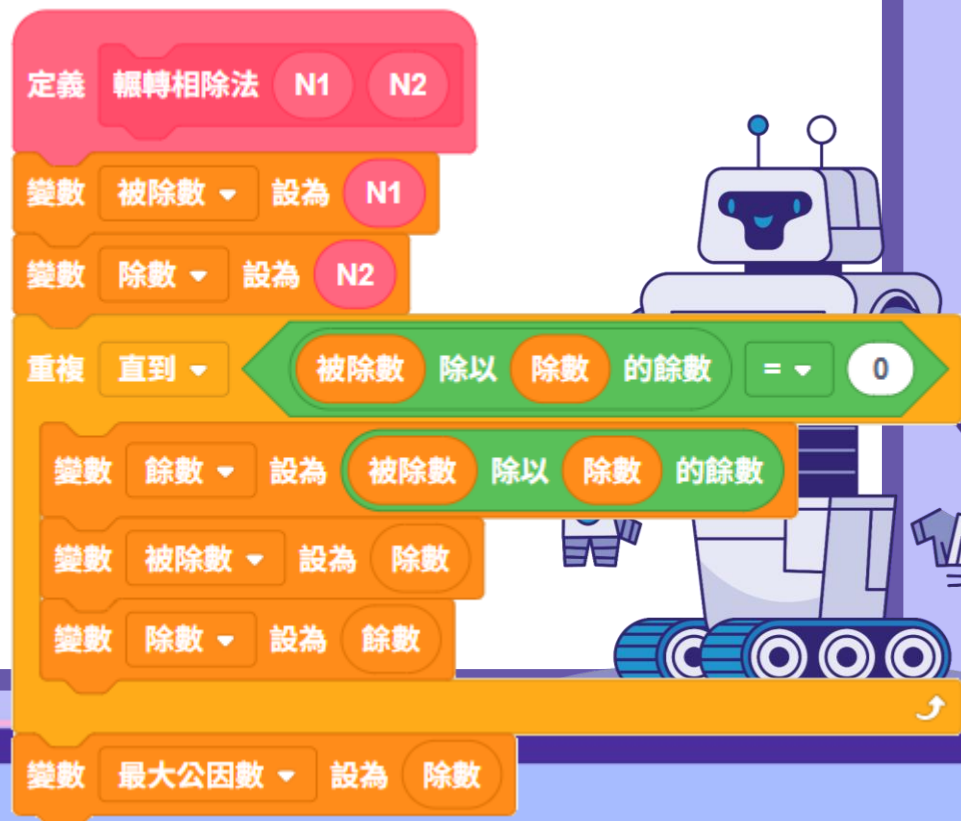
輸入

處理

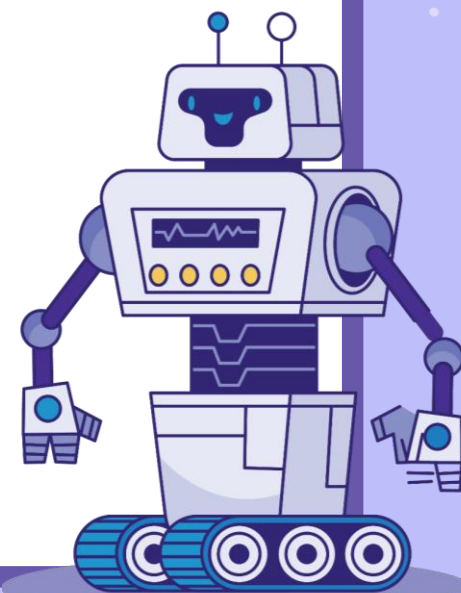
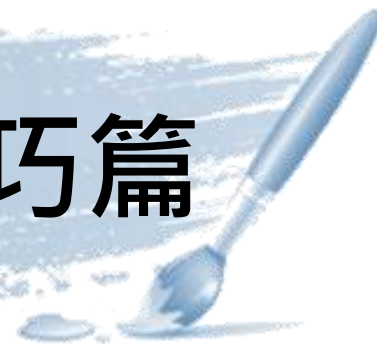
輸出



輾轉相除法



程式解題實戰技巧篇



卡關

- CheckList
 - 題意理解是否正確
 - 輸入、輸出格式是否正確(複製文字)
 - 數字、字串格式是否正確(格式轉換-0)
 - 初始值(數字、字串)
- Degug
 - 分段設定錨點輸出，確認是否正確

