

Tiny BASIC 解譯器

擴充版書面報告

組員名單：

人工二 B0928012 王晟翰

資工二 B0929027 梁祐愷

資工二 B0929049 向芸萱

西元 2022 年 6 月

目錄

壹、	系統發展	2
貳、	軟硬體執行平台及發展環境	2
參、	擴充功能	3
肆、	展示擴充功能的十隻程式	6
伍、	結論	11
陸、	參考文獻	11

壹、系統發展

一、發展目標

我們預計擴充 `gosub` & `return`、`if...then...else`、`load`、`save`、`shift-right`、`shift-left`、`XOR` 還有 24 個 `math function` 的功能。透過在 `reserved` 中新增 `gosub`、`return`、`load`、`save` 作為 `excuteTokens` 的判斷，再跳到對應的 `function` 中執行；還有再 `lex` 中新增了 `math function` 與 `else` 的定義，`math function` 在之後的 `solveExpression` 中進行判斷與執行，`else` 則是在 `ifHandler` 中讓程式當作判斷 `else` 的位置；最後則是在 `operators` 中新增 `<<`、`>>`、`^^`，在 `excuteTokens` 中判斷是否為 `OP` 後選取對應的 `operator` 執行。

二、發展人員

我們的分工是大家一起討論要修改的部分與方向並提供意見和改善的方法，主要是梁祐愷負責打 `code`，王晟翰負責找資料，向芸萱負責打報告，大家一起合力完成這份期末作業。

貳、軟硬體執行平台及發展環境

一、發展環境

我們的發展環境是利用 `Visual Studio Code` 和 `Python3.10`。

二、執行環境

引用 `python` 的 `math` 函數。

參、擴充功能

一、簡介

我們新增的擴充功能有 GOSUB RETURN (有巢狀)、IF THEN ELSE、存檔及讀檔，Operators 的部分有 Left、Right Shift 以及 XOR，還有新增了 24 個運算函式。

二、更動程式碼與函數

在原本的 reserved 裡面，我們增加了"SAVE", "LOAD", "GOSUB", "RETURN"；operators 裡面，在原本第四優先順位之前多新增一個順位有">>", "<<"兩個運算元、第二優先順位中多了"^" (我們自己發明的，代表 XOR)。

在分割跟定義 command 的 lex 函式中多了兩個判斷，第一個是判斷是否在 mathfunction，並定義為"MATHF"；第二個是判斷"ELSE"，並定義為"ELSE"。

在 executeTokens 的函式中，多判斷了我們新增的 4 個運算碼 "SAVE", "LOAD", "GOSUB", "RETURN"。

在 ifHandler 的函式中，我們多一個變數 elsePos，紀錄 else 的位置，並增加一個 for 迴圈去找 else 的位置，執行並判斷條件，當條件成立時做 THEN 到 ELSE 前的指令，不成立做 ELSE 後的指令。

在 solveExpression 的函式中，一開始除了原本的"OP", "NUM", "STRING", "ID" 還多判斷我們定義的"MATHF"讓他判斷。

三、新增程式碼與函數

首先，我們有引入 math 函式庫，並新增一個 mathfunction 列表，裡面是儲存我們的數學函數，以及多了一個 returnLine 來儲存 GOSUB 前的 linePointer。

我們增加四個函式，分別是用來存檔的 saveFunction 和用來讀檔的 loadFunction，以及用來執行 GOSUB 和 RETURN 的 gosubFunction 和 returnfunction。

四、實作細節說明

`lex(line)`是用來分割 `command` 並且將每個 `token` 定義，在定義中我們新增了 `MATHF` 與 `ELSE`，用來之後方便程式判斷 `token` 的屬性。

`executeTokens(tokens)`是用來判斷 `opcode` 的類型並執行，由於我們要做的 `SAVE`、`LOAD`、`GOSUB`、`RETURN` 皆屬於 `opcode` 的一種，所以在程式中新增這四種 `opcode` 的判斷。當讀到對應的 `opcode` 時，將 `opcode` 後剩餘的 `tokens` 放入對應的 `Function(saveFunction、loadFunction、gosubFunction、returnFunction)` 中執行。

在 `saveFunction` 中，剩餘的 `tokens` 代表要存入檔案的 `path`，開啟檔案位置並將屬性設為 `w`，`f=open(path, "w")`，接著讀取每一行程式，將程式透過 `f.write` 把 `line` 寫入 `path` 中直到全部寫入 `path` 的檔案中，最後再把路徑關閉 `f.close`，程式執行成功回傳 `true`。

在 `loadFunction` 中，剩餘的 `tokens` 代表要讀取檔案的 `path`，開啟檔案位置並將屬性設為 `r`，`f=open(path, "r")`，接著讀取檔案中每一行不是空白的程式並且在 `lex(line)` 中分割儲存，再放到 `executeTokens` 中執行，最後把路徑關閉 `f.close`，程式執行成功回傳 `true`。

在 `gosubFunction` 中，剩餘的 `tokens` 代表要跳入的副程式行號，先放入 `executeTokens` 執行，之後判斷型態是否正確，如果執行後的行號有在 `lines` 中，先將現在的 `linePointer` 放到 `returnLine` 的最後面，再將 `linePointer` 設為跳入的行號-1，程式執行成功回傳 `true`。

在 `returnFunction` 中，如果 `returnLine` 不為 0，將 `linePointer` 設為 `returnLine` 中的最後一個位置，再把 `returnLine` 中的最後一個值 `pop` 掉，程式執行成功回傳 `true`。

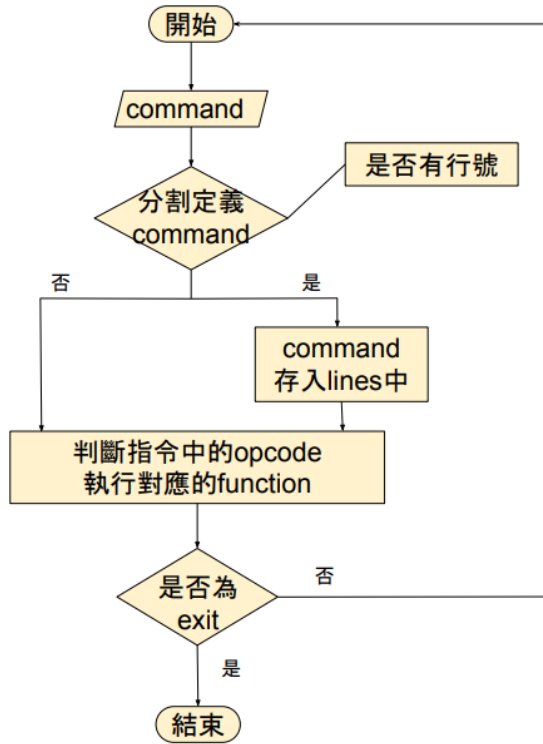
在 `ifHandler` 中，利用 `for` 迴圈找到 `ELSE` 的位置存入 `elsePos`，執行 `THEN` 前的 `if` 條件句，若結果為真，執行 `THEN` 後到 `ELSE` 前的程式；若結果為假，執行 `ELSE` 後的程式，程式執行成功回傳 `true`。

在 `solveExpression` 中，新增了 `math function` 的運算先判斷輸入的 `tokens` 類別是否為 `"OP"`、`"NUM"`、`"STRING"`、`"ID"`、`"MATHF"`，不是的話回傳 `None`。若 `tokens` 的類別為 `"MATHF"`，將剩餘的 `tokens` 執行相對應的數學函數，有 `"SIN"`、`"COS"`、`"TAN"`、`"ASIN"`、`"ACOS"`、`"ATAN"`、`"SINH"`、`"COSH"`、`"TANH"`、`"ASINH"`、`"ACOSH"`、`"ATANH"`、`"FLOOR"`、`"CEIL"`、`"ROUND"`、`"ABS"`、`"DEG"`、`"RAD"`、`"EXP"`、`"LN"`、`"LOG10"`、`"LOG2"`、`"SQRT"`、`"FACT"`，程式執行成功回傳運算後的答案。

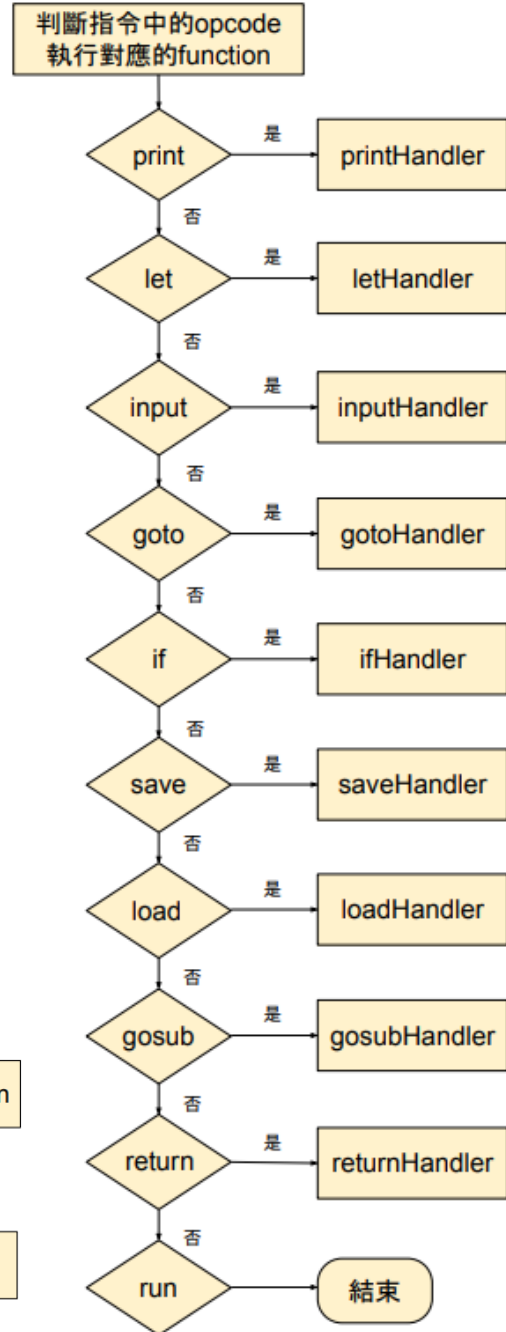
最後，在 `solveExpression` 中還有新增了三個 `operator`，分別是 `">>"`、`"<<"`、`"^^"`，若 `tokens` 的類別為 `"OP"`，將剩餘的 `tokens` 執行相對應的 `operator`，`operator` 的左或右 `expression` 如果不存在則程式回傳 `None`，程式執行成功回傳運算後的答案。

五、 流程圖說明

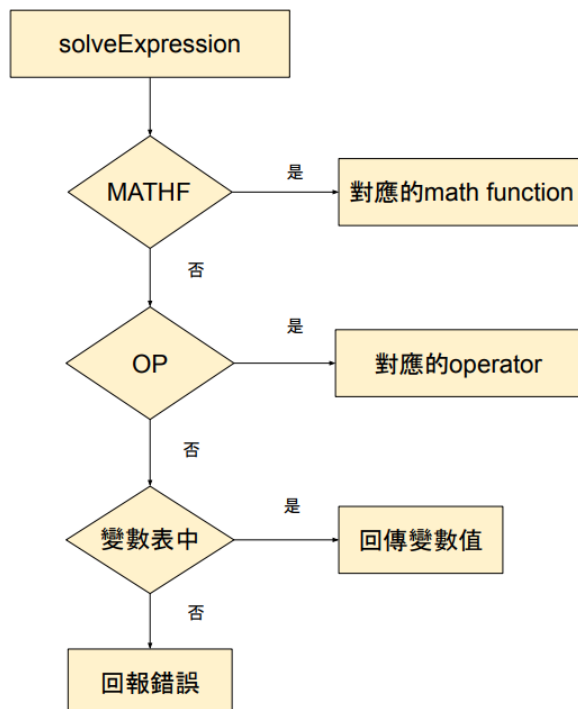
▼ 整體大致流程



▼ excuteTokens 中的流程



▼ solveExpression 中的流程



肆、展示擴充功能的十支程式

一、 GOSUB & RETURN

步驟及結果如下：

```
10 PRINT "HELLO"  
20 GOSUB 50  
30 PRINT "DAY"  
40 END  
50 PRINT "HAPPY"  
60 RETURN  
RUN  
HELLO  
HAPPY  
DAY
```

```
10 PRINT 1  
20 GOSUB 50  
30 PRINT 4  
40 END  
50 PRINT 2  
60 GOSUB 80  
70 RETURN  
80 PRINT 3  
90 RETURN  
RUN  
1  
2  
3  
4
```

二、 IF THEN ELSE

步驟及結果如下：

```
10 LET x = 0  
20 IF x > 1 THEN LET y = 100 ELSE LET y = -100  
30 PRINT y  
RUN  
-100
```

三、 Operators

- Left shift (<<) 、 Right shift (>>)

步驟及結果如下：

- ✧ Right shift

```
10 LET x = 10
20 LET y = -10
30 PRINT x >> 1
40 PRINT x >> 2
50 PRINT y >> 1
60 PRINT y >> 2
RUN
5
2
-5
-3
```

- ✧ Left shift

```
10 LET x = 10
20 LET y = -10
30 PRINT x << 1
40 PRINT x << 2
50 PRINT y << 1
60 PRINT y << 2
RUN
20
40
-20
-40
```

- XOR (^)

步驟及結果如下：

```
10 LET x = 2
20 LET y = 3
30 PRINT x ^ y
RUN
1
```


四、 存檔及讀檔

- 存檔

步驟及結果如下：

```
10 LET X = 1
20 LET Y = 2
30 PRINT X + 1
40 PRINT Y + 2
50 SAVE file.py
RUN
2
4
```

- 讀檔

步驟及結果如下：

```
10 LOAD file.py
RUN
2
4
OK.
LIST
10 LET X = 1
20 LET Y = 2
30 PRINT X + 1
40 PRINT Y + 2
```

五、數學運算函式

■ 數字運算

- "FLOOR" 無條件捨去
- "CEIL" 無條件進位
- "ROUND" 四捨五入
- "ABS" 取絕對值
- "SQRT" 開根號
- "FACT" 階層
- "DEG" 轉角度
- "RAD" 轉弧度
- "EXP" 以 e 為底的 x 次方
- "LN" 以 e 為底取 log
- "LOG10" 以 10 為底取 log
- "LOG2" 以 2 為底取 log

步驟及結果如下：

```
10 PRINT FLOOR 3.5
20 PRINT CEIL 3.5
30 PRINT ROUND 3.5
40 PRINT ABS -1
50 PRINT SQRT 100
60 PRINT FACT 5
70 PRINT DEG 1
80 PRINT RAD 60
90 PRINT EXP 2
100 PRINT LN 10
110 PRINT LOG10 2
120 PRINT LOG2 8
RUN
```

```
3
4
4
1
10
120
57.29577951308232
1.0471975511965976
7.38905609893065
2.302585092994046
0.3010299956639812
3
```

■ 三角函數

- "SIN" 正弦值
- "COS" 餘弦值
- "TAN" 正切值
- "ASIN" 反正弦值 ($-\pi/2 \sim \pi/2$)
- "ACOS" 反餘弦值 ($0 \sim \pi$)
- "ATAN" 反正切值 ($-\pi \sim \pi$)
- "SINH" 雙曲正弦值
- "COSH" 雙曲餘弦值
- "TANH" 雙曲正切值
- "ASINH" 反雙曲正弦值
- "ACOSH" 反雙曲餘弦值
- "ATANH" 反雙曲正切值

步驟及結果如下：

```
10 LET x = 0
20 PRINT SIN 30
30 PRINT COS 60
40 PRINT TAN 45
50 PRINT ASIN 0.5
60 PRINT ACOS 0.5
70 PRINT ATAN 1
80 PRINT SINH x
90 PRINT COSH x
100 PRINT TANH x
110 PRINT ASINH 0
120 PRINT ACOSH 1
130 PRINT ATANH 0
RUN
```

0.5
0.5
1
30
60
45
0
1
0
0
0
0

伍、結論

做 Tiny Basic 的過程中，首先我們要把老師預先給的程式理解透徹，例如每一段的條件式的成立條件，以及它的作用。在系統程式課本上所學的程式，通常都是一個段落，或者是一個部分，而且都是非常淺顯易懂的。相較起來，這是我們第一次看到比較完整且複雜的程式。一開始，我們都覺得程式碼彷彿天書一般，但靜下心來，一行一行去了解或者互相討論，發現其實並沒有想像中的困難。

我們去做擴充功能所採取的戰術是先把我們認為能做出來的完成，像是 left shift、right shift，這個在計算機概論和計算機組織中都有學過，很快就可以把之前所學，套用在 tiny basic 上。For 迴圈原本認為並不會太難，畢竟我們時時刻刻都會用得到，但實際做的時候，困難重重，因此選擇先完成 math function。Math function 有些的基礎功能，可以從 python 去做引用，稍作修改，就可以很好的執行在 tiny basic 上。有些則是透過去了解要做的 math function 的數學基本定理，慢慢打出來。

做擴充功能時，我們其實遇到過許多問題，像在 abs(取絕對值)我們遇到一個問題，原本我們無法在 abs 中先做運算再取絕對值，我們花了好幾天修改，慢慢討論才解決這個問題。Sin 和 Cos 剛開始我們發現出來的值都不正常，後來才發現 python 預設是使用 rad(弧度)，所以為了不讓使用者不好輸入，我們特地把弧度和角度的功能都新增給使用者，讓使用者能自己選擇，而不用預先做轉換。XOR 代號上的選擇，因為使用 XOR 會被判定為變數，我們把要用 XOR 的代號改成 "^"，除此之外我們幾乎完成大部分老師要求的功能。

透過這次 Tiny basic 專題，把原本課程上的理論知識很好轉換成實際運用，實際運用對於資工和 AI 學程都是非常重要的技能，專題的活動很好的提升我們這項技能。

陸、參考文獻

Math --- 數學函式—Python 3.10.5 說明文件

<https://docs.python.org/zh-tw/3/library/math.html>

负数在 python 中的右移计算 | Python 笔记

<https://www.pynote.net/archives/3761>

[JAVA]運算子-運算子的優先順序

[https://pclevinblog.pixnet.net/blog/post/314563342-](https://pclevinblog.pixnet.net/blog/post/314563342-%5Bjava%5D%E9%81%8B%E7%AE%97%E5%AD%90-%E9%81%8B%E7%AE%97%E5%AD%90%E7%9A%84%E5%84%AA%E5%85%88%E9%A0%86%E5%BA%8F)

[%](https://pclevinblog.pixnet.net/blog/post/314563342-%5Bjava%5D%E9%81%8B%E7%AE%97%E5%AD%90-%E9%81%8B%E7%AE%97%E5%AD%90%E7%9A%84%E5%84%AA%E5%85%88%E9%A0%86%E5%BA%8F)

[%](https://pclevinblog.pixnet.net/blog/post/314563342-%5Bjava%5D%E9%81%8B%E7%AE%97%E5%AD%90-%E9%81%8B%E7%AE%97%E5%AD%90%E7%9A%84%E5%84%AA%E5%85%88%E9%A0%86%E5%BA%8F)

[%](https://pclevinblog.pixnet.net/blog/post/314563342-%5Bjava%5D%E9%81%8B%E7%AE%97%E5%AD%90-%E9%81%8B%E7%AE%97%E5%AD%90%E7%9A%84%E5%84%AA%E5%85%88%E9%A0%86%E5%BA%8F)