**關於C subset + type check rules的描述**

#後面有單獨 **type check rules**

1. **關鍵字**

Return: 後面可以接 算術運算、string、單個字元

1. **資料型態 (宣告變數):**

注意變數不能是key word (printf int…，包括‘c’,’d’,’f’,’s’也是key word)

int, float, double, short: 這邊一定要給一個ID，初始化或是不初始化都可以，但一次只能定義一個變數。

char : 可以定義變數為單個字元，或是定義一個字串(定義字串時的格式為 char name[] = “your\_string”;)

1. **註解方式:**

單行註解: //這行在雙斜線後的文字註解

多行註解: /\*中間是註解內容\*/

1. **program statement相關之邏輯、數學運算子**

加一減一: ++、-- (但是只有後綴可以用)

加減乘除餘: +、-、\*、/、%

比較運算: <、>、<=、>=、==、!=

賦值運算: =、+=、-=、\*=、/=

1. **其他標點符號**

分號: ;

小括號: ()

中括號:[]

大括號:{}

逗號: ,

井字號: #

問號: ?

冒號: :

點號: .

底線: \_

1. **支援之程式控制結構**

**條件判斷:**

**If、else :**

* if , else if後面一定要有小括號，但是後面如果是單一個statement那就不一定要大括號。
* 最後一個else一定要有一個大括號將後面的statements給包起來。
* If condition 部分只能有boolean的type，如果使用者輸入是arith\_expression、string、char，會有 Error! Type error的警告。
* 如果If condition 部分的回傳type是Error的話會跳出 Error! condition section compare error.。
* Else if condition部分也是沿用if的判斷方式。
* If會有一個 typeInfo回傳if,else if的condition部分是否正確，如果正確的話回傳Void，不正確回傳Error，這邊不考慮if else結構中的statement的正確性。

**迴圈:**

* **for、while、break、continue:**

**for迴圈的小括號中有三個區域 (區域1; 區域2; 區域3):**

1. **區域1:**
   * 可以在這邊定義 (int, float, double, short) 型態的變數。
   * 可以給任意 ID 賦值 (這邊是用算術運算，所以也不一定只能是單純一個數字)。
   * 也可以單純定義一個 string, char。
   * 如果定義時有錯誤會列印Error init section error。
2. **區域2:**
   * 中間是一個比較運算(回傳boolean type)，用來確認 for 迴圈要不要繼續執行下去。
   * 比較運算是由一個 ID 與 (ID, int, float) 其中一個比較。
   * 如果出現arith\_expression,string,char，會列印 Error for loop condition section type error。
   * 如果比較運算有問題會列印，Error for loop condition section compare error.。
3. **區域3:**
   * 算術運算區域，可以執行最常用的 i++, i--- 等。
   * 如果算術運算有問題，會列印Error for loop increase section error.

* 如果for loop小括號中的三個區域有任意一個有問題，那回傳時的type就會是Error，都正確則是回傳Void。
* For 之後的statements一定要有大括號刮起來。

**While :**

* 後面必定接一個小括號，小括號裡面要有一個比較運算或是算術運算，後面如果是接單個 statement 就不一定要有大括號，如果是多行程式就要用大括號括起來。
* while condition 部分只能有boolean的type，如果使用者輸入是arith\_expression、string、char，會有 Error! Type error的警告。
* 如果condition 中的compare有問題，會列印出Error while condition section compare error.。
* 如果都沒有問題的話回傳的type(變數typeInfo)會是 TypeInfo.Void。
* 上述型態or格式判斷如果有問題的話會是回傳TypeInfo.Error。

1. **program execution**

程式只能有一個main function，main function回傳值可以是(int, float, char, double, void)。

可以有for、while但要符合前一段的結構。

變數名稱不能是keyword。

我們有支援基本的算術運算

* 算術運算

這邊我們一層層介紹算術運算的運作方式與報錯方式。

1. arith\_expression:加減的運算，如果再加減的兩邊的資料type不同的話會顯示 Error Type mismatch for the operator.
2. multExpr:乘除的運算，如果兩邊的資料型別不同的話會顯示，Error Type mismatch for the operator.。
3. signExpr:加上負號的操作
4. double\_add\_sub:’++’and’--’只有後綴可以用，如果資料不是Integer的話就會跳Error! need Integer type primaryExpr。
5. primaryExpr:運算基本單元，可以是整數，浮點數，還有ID。

如果ID的格式不是Integer,Float的話會跳Error ID not Int or Float.。

有支援include library

1. **部分函式**

* printf、scanf、main

printf、scanf:

* 可以有一個或是兩個參數，但是參數(限定%d, %f,%c,%s)以外的字元不能出現在雙引號區域中，然後前面有一個參數會面就必須有一個變數來接輸入或是給輸出。
* 如果輸入的變數是沒有定義過的會列印出，"Error! printf (first/second) var undeclared identifier." (first/second)是看是第幾個變數出現問題。
* 如果前面的(%d, %f,%c,%s)與後面給的變數type不相同的話也會跳出錯誤訊息，錯誤訊息中會說明是第幾個變數有問題以及正確的type是什麼，例如("Error! "+": first var type is wrong, need integer.)。

**type check rules:**

* **基本規則:**
* 變數在使用前必須先宣告or定義
* 不能重複宣告
* Operands兩邊的type必須相同
* Assignment兩邊的type必須相同
* Compare 回傳type為boolean
* **較詳細的說明和剩餘的小規則:**

**條件判斷:**

**If、else :**

* If condition 部分只能有boolean的type，如果使用者輸入是arith\_expression、string、char，會有 Error! Type error的警告。
* 如果If condition 部分的回傳type是Error的話會跳出 Error! condition section compare error.。
* Else if condition部分也是沿用if的判斷方式。
* If會有一個 typeInfo回傳if,else if的condition部分是否正確，如果正確的話回傳Void，不正確回傳Error，這邊不考慮if else結構中的statement的正確性。

**迴圈:**

* **for、while、break、continue:**

**for迴圈的小括號中有三個區域 (區域1; 區域2; 區域3):**

1. **區域1:**
   * 如果定義時有錯誤會列印Error init section error。
2. **區域2:**
   * 中間是一個比較運算(回傳boolean type)，用來確認 for 迴圈要不要繼續執行下去。
   * 比較運算是由一個 ID 與 (ID, int, float) 其中一個比較。
   * 如果出現arith\_expression,string,char，會列印 Error for loop condition section type error。
   * 如果比較運算有問題會列印，Error for loop condition section compare error.。
3. **區域3:**
   * 如果算術運算有問題，會列印Error for loop increase section error.

* 如果for loop小括號中的三個區域有任意一個有問題，那回傳時的type就會是Error，都正確則是回傳Void。

**While :**

* while condition 部分只能有boolean的type，如果使用者輸入是arith\_expression、string、char，會有 Error! Type error的警告。
* 如果condition 中的compare有問題，會列印出Error while condition section compare error.。
* 如果都沒有問題的話回傳的type(變數typeInfo)會是 TypeInfo.Void。
* 上述型態or格式判斷如果有問題的話會是回傳TypeInfo.Error。

**算術運算**

這邊我們一層層介紹算術運算的運作方式與報錯方式。

1. arith\_expression:加減的運算，如果再加減的兩邊的資料type不同的話會顯示 Error Type mismatch for the operator.
2. multExpr:乘除的運算，如果兩邊的資料型別不同的話會顯示，Error Type mismatch for the operator.。
3. signExpr:加上負號的操作
4. double\_add\_sub:’++’and’--’只有後綴可以用，如果資料不是Integer的話就會跳Error! need Integer type primaryExpr。
5. primaryExpr:運算基本單元，可以是整數，浮點數，還有ID。

如果ID的格式不是Integer,Float的話會跳Error ID not Int or Float.。

**部分函式**

* printf、scanf、main

printf、scanf:

* 可以有一個或是兩個參數，但是參數(限定%d, %f,%c,%s)以外的字元不能出現在雙引號區域中，然後前面有一個參數會面就必須有一個變數來接輸入或是給輸出。
* 如果輸入的變數是沒有定義過的會列印出，"Error! printf (first/second) var undeclared identifier." (first/second)是看是第幾個變數出現問題。

如果前面的(%d, %f,%c,%s)與後面給的變數type不相同的話也會跳出錯誤訊息，錯誤訊息中會說明是第幾個變數有問題以及正確的type是什麼，例如("Error! "+": first var type is wrong, need integer.)。