

# 10410EE231001 Introduction to Programming

## Assignment 1

**Due: Nov 27th, 2016 (11:59pm)**

### Instructions

1. Your program should be compilable with Code::Blocks and executable on a Windows platform (PC).
2. Submit your project or your source file (main.c) to the iLms system: <http://lms.nthu.edu.tw>. Your code should be well-commented so that the TAs can easily understand. With proper comments, even if your program cannot be compiled, you may still get partial credits.
3. Please name your file as student ID.c (Ex: 103064533.c)
4. Late submission will incur 10% penalty per day up to 3 days. After that, assignment submission will be closed and no submissions will be accepted.

### Compiler and Calculator

In this assignment, your program should have two sub-functions. One is **compiler**, the other is **calculator**. You need to use an option to separate two component. If type '1' the program executes the **compiler**, and type '2' the program executes the **calculator**. The following will explain the detail of two sub-functions.

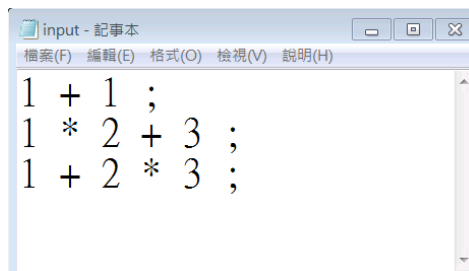
#### Compiler

Step1: Open and read a text file.

Step2: Compile the content of the file, if there is no error, transform the file to the hexadecimal format. Otherwise, print the error line.

Step3: Print the result.

The text file will be operators( + , - , \* , / ) and numbers (0~9), such as



```
input - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
1 + 1 ;
1 * 2 + 3 ;
1 + 2 * 3 ;
```

There is a white-space ' ' between characters and there is a semicolon ';' at the end of each expression.

The hexadecimal format is illustrated as the following tables.

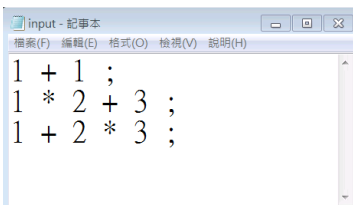
Character	Decimal	Hexadecimal
0	0	00
1	1	01
2	2	02
3	3	03
4	4	04
5	5	05
6	6	06
7	7	07
8	8	08
9	9	09

Character	Decimal	Hexadecimal
+	10	0A
-	11	0B
*	12	0C
/	13	0D
;	14	0E

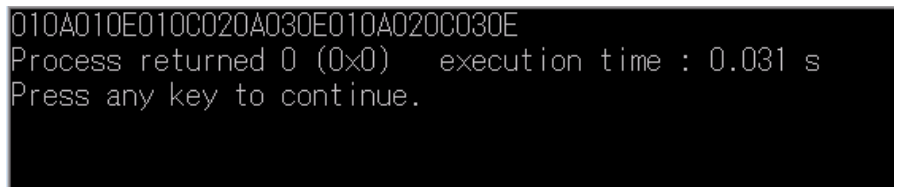
For example: '1 + 1 ;' will be '010A010E' in the hexadecimal format.

If the expression in the text file has an **error**, you should print the error message and stop compiler immediately. The input cannot have more than two operators, so '1 + 1 + 1 + 1 ;' will be an error. The answer line will have at most 100 characters.

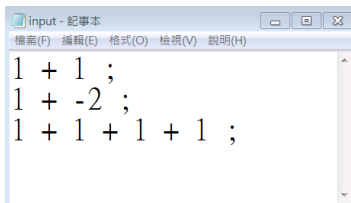
Sample input 1:



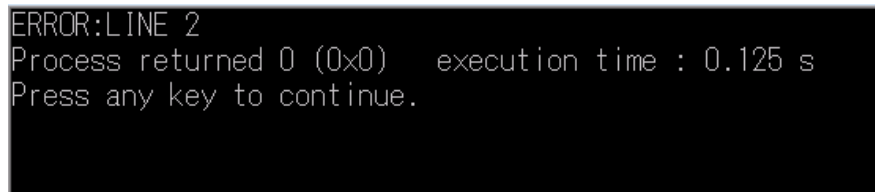
Sample output 1:



Sample input 2 :



Sample output 2 :



**What you should pay attention to**

1. The transformation result DOES NOT have the newline character '\n'.
2. In sample input 2, although line 1 is correct, you should not print its results since it cannot compile correctly. Moreover, both line 2 and line 3 have errors, but you should not show that line 3's errors. When you find that line 2 has error, you must stop compiler immediately!

3. We suggest you use `fscanf( ,"%c", )`.
4. Please check the input very carefully, as there are many kinds of errors!

Some error samples ( ):

- `1 + + 2 ;` (two contiguous operators)
- `1 + 1` (no semicolon ‘;’)
- `11 + 2 ;` (two contiguous numbers)
- `1 1 + 2 ;` (two contiguous numbers)
- `-1 + 1 ;` (negative sign alone before a number)
- `1 + 1 + 1 + 1 ;` (more than two operators)
- `1 % 2 ;` (not +, -, \* or /)
- `9 + 1 ; 1 + 1 ;` (two semicolons ‘;’ in the same line)
- `a + b - c ;` (not numbers)

## Calculator

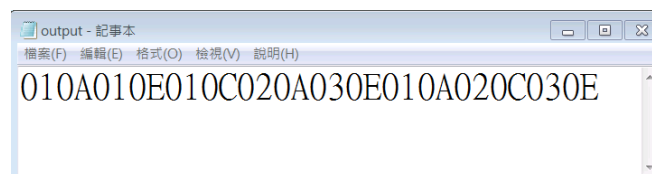
Step1: Open and read a text file. The content in the text file should already be in hexadecimal format.

Step2: Decode the text file to decimal format or characters (depending on your code).

Step3: The decoded file must compile by your **Compiler** to check whether or not it can be compiled.

Step4: If it cannot be compiled, print the error line. Otherwise, calculate and print the result.

The text file will be like:



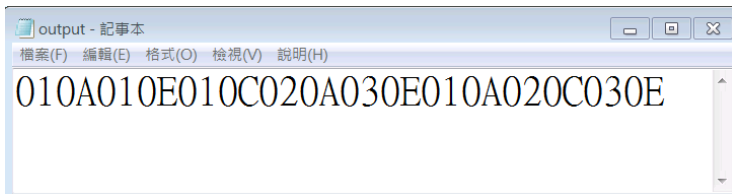
Please use the previous table to decode the text file and calculate the result.

After decoding, you must compile it by your **Compiler**. If it cannot be compiled, you must show the error message and end your program.

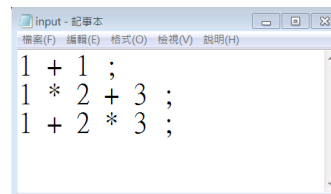
If it can be compiled, you still need to check whether each formula has division-by-zero. If it has, please show ‘ERROR’ and continue to calculate the next formula.

The input will always be divisible. That is, you only need to declare type of int, and please follow the priority of four fundamental operations of arithmetic.

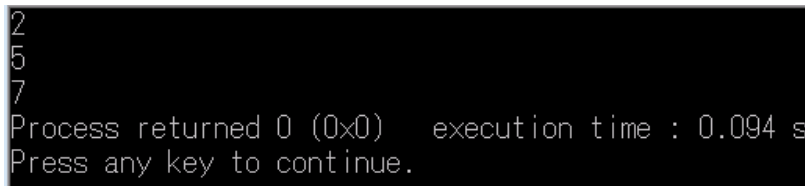
### Sample input 1:



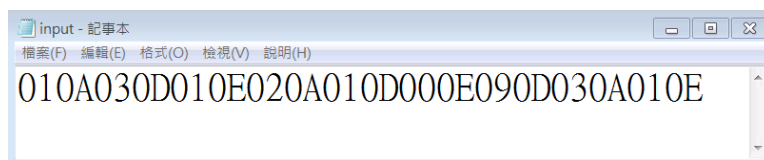
### Decode:



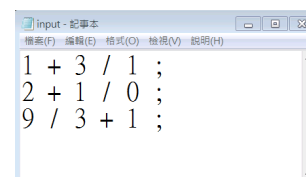
### Sample output 1:



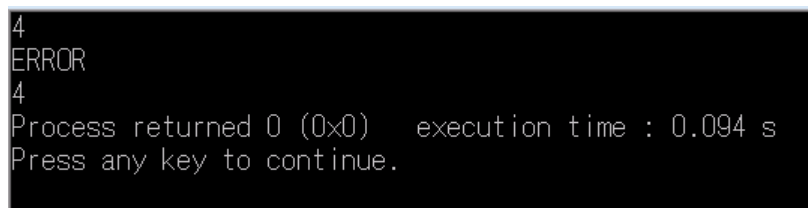
### Sample input 2:



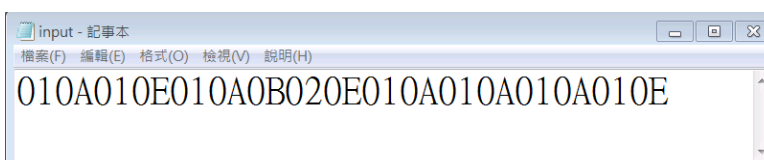
### Decode:



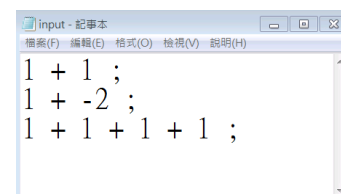
### Sample output 2:



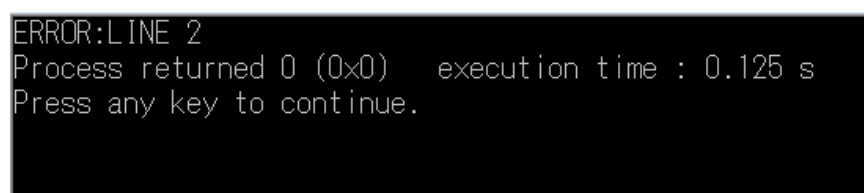
### Sample input 3:



### Decode



### Sample output 3:



### **What you need to pay attention to**

1. You DO NOT need to show the result of decode. TA just show you the correct formula after you decode the text file.
2. Remember to add a newline character '\n' after you calculate each formula.

### **Guidelines**

1. Mark weightings: correctness 70%, Source code readability 30%.
2. Correctness: Make sure you understand what the program should do in every case, including special cases.
3. Program style:
  - A. Your program should include at least 6 functions (including **Compiler** and **Calculator**).
  - B. Their functionality should be well-defined, easily understandable, and clearly documented as comments within the source code. Add sufficient and appropriate comments to your program.
4. You are welcome to discuss with each other, but DO NOT COPY OTHER PEOPLE'S WORK. Plagiarism is a serious offense. Not only will you get no points in this assignment, but you may also be reported to the university.