

# Assignment I — Structs and Arrays

TA: Nick (weihaan9453@gmail.com)

Deadline: 2017 Oct. 17, 11:59pm

## Structs and Arrays

You are a new programmer and work for an IT company. The Institute of Astronomy's researchers need to spend a lot of time calculating on many equations, for example, light-year conversion, solar eclipse, temporal and spatial variation, etc. Therefore, they want to hire your IT company to make a small program. The program should be able to calculate two very large numbers that can't be stored in the integer type. Your boss is confident that you can solve this problem.

We all know that the variables of various programming languages have the upper limit. A typical integer holds either  $2^{16}$  or  $2^{32}$ . However, in some cases, we have to deal with a fairly large number of integers. We assume that there are no other data type (e.g., float and double) available. To settle this problem, we can divide the numbers individually into separate digits firstly. Then, we declare a char array to store these digits. Finally, each pair of digits can be calculated separately.

## Action menu

Because the program is meant to be used by a user, who does not want to modify source code, please make your program user-friendly. You must add an action menu. When the program is being executed, a question with multiple options will be shown. The Figure1 is an example for the action menu:

```
What do you want to do?  
  
1) Read numbers from a file  
2) Show the input numbers  
3) Write numbers to a file  
4) Calculate the big numbers  
0) Exit  
  
>
```

Figure 1. An Illustration of an action menu.

The user enters the desired option (i.e., 0, 1, 2, 3 or 4) and the program executes the function. For an action menu, you can use this piece of code as a template. The following sections explain the detail for each function.

```
int flag=1;
while(flag){
    printf("Choose action:\n1. Do stuff\n2. Do other stuff\n0. Exit");
    int number;
    scanf("%d", &number); // User input
    switch(number){
        case 1:
            /* Do stuff */
            break;
        case 2:
            /* Do other stuff */
            break;
        case 0:
            /* Exit */
            flag=0;
            printf("bye-bye!\n");
    }
}
```

## 1. Read numbers from a file

Because the researchers are inured to record the digits in the document, please write a program that can read numbers from a file and declare a char array to store the numbers into a defined structure. The foregoing process is shown in Figure 2.

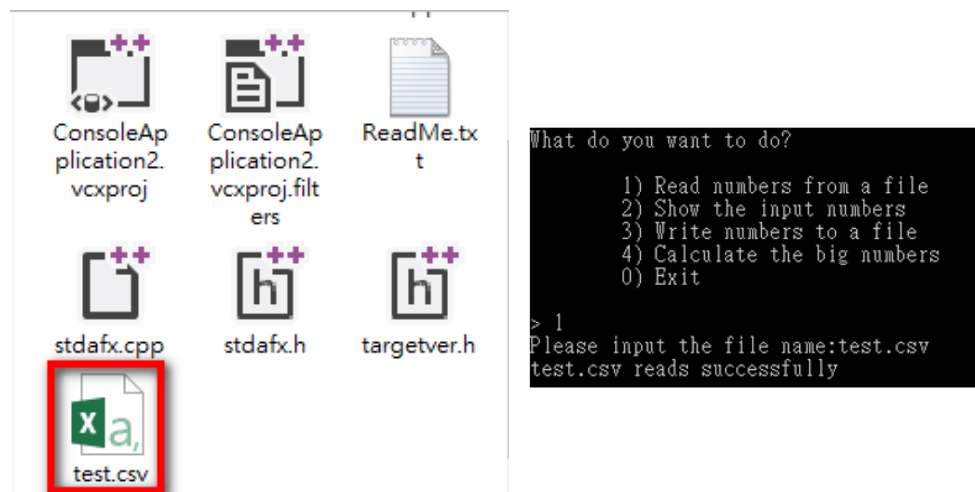


Figure 2. An Illustration of reading a file.

Your program should be able to read and save multiple pairs of large numbers into an array. To make the records better usable for the researchers, we want to allow using third party products, such as Microsoft Excel or LibreOffice Calc to work with our program. Luckily, these programs can read file in CSV format. Please let your program operate with CSV files. Your program should read a CSV file that modifies numbers into scientific notation, and the output numbers should also follow the CSV format. In our tests, you can expect that there will be exactly 10 pairs of large numbers. The records will be given in the csv-format as below.

ID, Number1, Number2

## 2. Show input numbers

After the first step is completed, the researchers don't know whether the numbers had read or not. So please program the function which can show the numbers as well. Because the numbers is big and lengthy, please express the numbers in scientific notation. In Figure 2, the screenshot on the left is a test .csv file which contains the 10 pairs of input numbers. The screenshot on the right shows the corresponding outputs in scientific notation.

ID	Number1	Number2
1	9	9
2	333	111
3	1.00E+17	1.00E+17
4	6.67E+14	1.11E+14
5	2147483647	2147483647
6	9000000000	4500000000
7	88888	6666
8	44464	8949
9	6.00E+38	3.00E+38
10	3.00E+43	1.00E+43

```
What do you want to do?
    1) Read numbers from a file
    2) Show the input numbers
    3) Write numbers to a file
    4) Calculate the big numbers
    0) Exit
> 2
Here is your records
No.1 : 9.00e+000 , 9.00e+000
No.2 : 3.33e+002 , 1.11e+002
No.3 : 1.00e+017 , 1.00e+017
No.4 : 6.67e+014 , 1.11e+014
No.5 : 2.15e+009 , 2.15e+009
No.6 : 9.00e+008 , 4.50e+008
No.7 : 8.89e+004 , 6.67e+003
No.8 : 4.45e+004 , 8.95e+003
No.9 : 6.00e+038 , 3.00e+038
No.10 : 3.00e+043 , 1.00e+043
```

Figure 2. An Illustration of showing read numbers.

## 3. Write numbers to a file

In this section, please program the function which can write numbers to a file. You have to read a CSV file first. Then, please input two numbers from the command line and write these two numbers to the CSV file as a new record. You must append the record to the end of the records in the file and increment the ID number. Figure3 illustrate adding two numbers to a file.

```

> 1
Please input the file name:test.csv
test.csv reads successfully

What do you want to do?

    1) Read numbers from a file
    2) Show the input numbers
    3) Write numbers to a file
    4) Calculate the big numbers
    0) Exit

> 3
No.11
Number1:99887766554433221100123456789
Number2:78945612309876543210741258963

```

ID	Number1	Number2
1	9	9
2	333	111
3	1.00E+17	1.00E+17
4	6.67E+14	1.11E+14
5	2.15E+09	2.15E+09
6	9E+08	4.5E+08
7	88888	6666
8	44464	8949
9	6.00E+38	3.00E+38
10	3.00E+43	1.00E+43
11	9.99E+28	7.89E+28

Figure 3. An Illustration of writing numbers to a file.

## 4. Calculate the big numbers

Please program four basic Mathematical Equations, namely addition, subtraction, multiplication, division so that a user can execute calculation on numbers. When the user choose (4) to calculate the numbers, please program an action menu first. After the user chooses the operator, it inquires the user whether to load the existing numbers by specifying an ID for a record or not. If you choose (1) and do not read a CSV file yet earlier, it will remind you to read a CSV file first. The above description is shown in Figure 4.

**ATTENTION:** You must output a result.txt that records the calculation results.

```

What do you want to do?

    1) Read numbers from a file
    2) Show the input numbers
    3) Write numbers to a file
    4) Calculate the big numbers
    0) Exit

> 4

=====
=
= Which operators do you want to choose?
= (+) Addition
= (-) Subtraction
= (*) Multiplication
= (/) Division
= (q) Quit
=
=====
+

=====
=
= Which items do you want to choose?
= (1) Load the existing numbers by ID
= (2) Input two numbers to calculate
= (0) Quit
=
=====
1
What index do you want to read ?

```

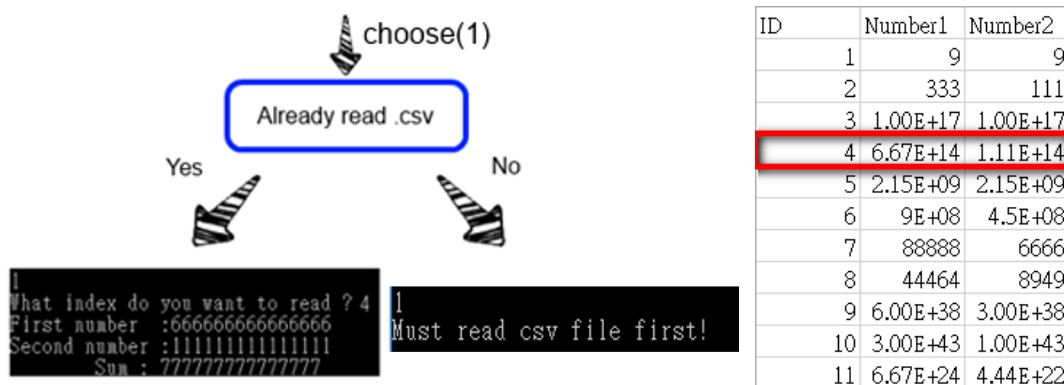


Figure 4. An Illustration of loading the existing numbers by specifying an ID number.

If you choose (2) Input two numbers to calculate, now you can add two numbers to calculate directly. Figure5 shows the corresponding output when choosing option (2).



