

CS2100 Computer Organisation
Lab 02: A little bit more of C (Week 31st August 2020)
Instruction

Short and clean

We have separated the lab information into i) **instruction** and ii) **report**. You **ONLY** need to submit your **report** into Luminus Folder no longer than 2359 on the same day you have the online lab. Whenever there is a question in the instruction (easily identified as they have **[X pts]** tagged to the end), write / type your answer in the corresponding location in the **report** printout.

Objective:

You will learn how to use arrays, functions and structures in C.

Preparation (before the lab):

1. Read up the relevant C syntax. [References: Lecture 04, 05 and 06]
2. Finish the report before coming to lab. It is hard to learn, code, debug within the ~50 minutes lab time.

Part a

Procedure:

1. Compile **lab2a.c**.
2. Observe the output. Modify the code such that the array element "2" is printed instead. Give the printf() statement in report. **[1 pt]**
3. Find out the purpose of the C operator **sizeof()**? Briefly describe in report. **[2 pt]**
4. Using **sizeof()**, add additional code to **lab2a.c** such that it prints out the **number of element** in the **ageArray**. Note: You cannot **hardcode** the output, i.e. the result must be calculated based on sizeof(). For example, if you add another element to the end of the ageArray, "int ageArray = {2, 15, 4, 21};", the output of your program is:

```
2          //from part (2), no change
4 array elements  //new message
```

[Hint: Think about the relationship between the entire array vs a single element]

Demonstrate your **completed lab2a.c** to the Lab TA. **[3 pts]**

Part b

Procedure:

1. **lab2b.c** uses the Fraction structure discussed in lecture and tutorial 2, but combined with array.
2. Complete the function **printFractionArray()**. Use the **printFraction()** function to simplify your code. The output should look like:

```
1 / 2, 3 / 4, 5 / 6,    //for simplicity, it is ok to have a dangling "," for the last fraction
```

Give the function body in your report. **[3 pts]**

3. Use similar idea from [part a, step 4] to answer the following:
 - a. Can you print out the number of fractions in **fArray[]** in **main()**?
 - b. Can you print out the number of fractions in **fArray[]** in **printFractionArray()**?

For case(s) that didn't work, briefly explain. **[3 pts]**

Marking Scheme: Report – 9 marks; Demonstration – 3 marks; Total: 12 marks.

Program lab2a.c

```
#include <stdio.h>

void display(int age)
{
    printf("%d\n", age);
}

int main()
{
    int ageArray[] = { 2, 15, 4 };

    display(ageArray[0]);
    printf("%d array elements\n", size);

    return 0;
}
```

Program lab2b.c

```
#include <stdio.h>

struct Fraction {
    int num;
    int den;
};

void printFraction( struct Fraction *fptr )
{
    printf( "%d / %d", fptr->num, fptr->den );

    //Can also be written as
    //printf( "%d / %d", (*fptr).num, (*fptr).den );
}

void printFractionArray( struct Fraction fArray[], int size)
{
    //Your work here
    //Dont panic: work out the syntax step-by-step
    //Hint: Need 3-4 lines of code only. Remember to reuse printFraction()
    function
}

int main()
{
    struct Fraction fArray[3] = { {1, 2}, {3, 4}, {5, 6} }; //an array of 3
    fractions

    printf("Fraction 1 = %d / %d\n", fArray[0].num, fArray[0].den); //observe
    how to combine the syntax of array and structure

    printFractionArray( fArray, 3 ); //Why do we pass in the array size?

    return 0;
}
```