CS2100: Computer Organisation
Lab #2: Debugging using GDB II

[This document is available on LumiNUS and module website http://www.comp.nus.edu.sg/~cs2100]
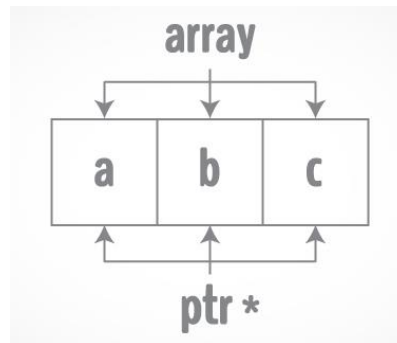
Name: _____   Student No.: _____

Lab Group: _____

C Arrays

Array is a kind of data structure that can store a <u>fixed-size</u> sequential collection of elements of the <u>same type</u>. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.


Instead of declaring individual variables, such as number0, number1... and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index which starts from 0.

All arrays consist of <u>contiguous memory locations</u>. The lowest address corresponds to the first element and the highest address to the last element.

C Functions and Arrays

In C programming, a single array element or an entire array can be passed to a function. A single value will be passed by value, whereas when passing the whole array, it is always passed as a reference to the first element of the array.
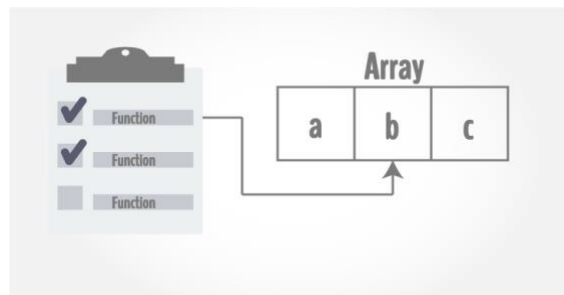
Objective:
You will learn how to use arrays and functions in C.

Preparation (before the lab):
Please refer to lab#1.

Procedure:

1. Locate the **lab2a.c** and **lab2b.c** files in the zip file that this lab document came in.

2. Compile **lab2a.c** with **gcc** using the following command: `gcc –o lab2a lab2a.c`

3. What is the output of the program? Can you change it to **"2"**?
    Note: The output should be related to the **ageArray** such as an element in **ageArray**.

    > 4. Change to display(ageArray[0]);

4. What is the purpose of the operator **sizeof**? What datatype will **sizeof** give **"1"** value for on all architectures?

    > sizeof will return the number of bytes of an object or type
    >
    > -char

5. Can you get the number of elements in **ageArray**? To produce the following output:

    2
    Size of the array is 4
    Modify the main function, write it below and show your labTA the output.
    Note: The output **"2"** and size of array (i.e., **4** (_four_)) should be related to **ageArray**
    such as an element in **ageArray** and the number of elements in **ageArray**.

    > printf("Size of the array is %d\n", ageArray[2]);
    >
    > printf("Size of the array is %ld\n", (sizeof(ageArray)/sizeof(ageArray[0])));

6. Compile **lab2b.c** with **gcc** using the following command: `gcc –o lab2b lab2b.c`

7. Can you give 2 ways of displaying the stored value and address value of the first element of an array?

   printf("Stored value is: %d\n, address value is %p\n", array[0], array)

   printf("Stored value is: %d\n, address value is %p\n", *array, &array[0])

8. Can you define the function **hexToDecimal(char hex[], size_t size)** in the lab2b.c <u>using pointers</u> to traverse the array? Write your function below and show your labTA the output.
   Note: You are not allowed to use **strtoul**, **strtol**, or other functions from **stdlib.h**.
   *Hint: Reading from the back of array is easier. Furthermore, you are already given the function **hexVal(char hex)** to simplify your work.*

   ```
   long hexToDecimal(char hex[], size_t size) {
       long total = 0;
       int counter = 0;
       for (int i = size; i > 0; i--) {
           total += (hexVal(*(hex+i-1)) * pow(16,counter));
           counter += 1;
       }
       return total;
   }
   ```

9. Why do we pass the size of the array to the **hexToDecimal** function in lab2b.c? Can we calculate the size of the array inside the function?

   To simplify the variables used in the function

   -can but means every new functions need to calculate again since variable would
   be local

10. What is the format specifier to print a variable of datatype **size_t**?

    %zu

Marking Scheme: Report – 5 marks; correct output – 5 marks; Total: 10 marks.

Program lab2a.c

```
#include <stdio.h>

void display(int);

int main() {
    int ageArray[] = { 2, 15, 4 };
    display(ageArray[2]);
    return 0;
}

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

Program lab2b.c

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int hexToDecimal(char[], size_t);
int hexVal(char);

int main(void) {
    char hex[8];
    size_t len;

    printf("Enter up to 7 hexadecimal characters: ");
    fgets(hex, 8, stdin);
    len = strlen(hex);

    /* End-of-Line Check */
    if(hex[len-1] == '\n') {
        len = len - 1;
        hex[len] = '\0';
    }

    printf("You entered: %s\n", hex);
    printf("The value in decimal is: %d\n", hexToDecimal(hex,
len));

    return 0;
}
```

```
int hexVal(char hex) {
      switch(toupper(hex)) {
            case '0': return 0;
            case '1': return 1;
            case '2': return 2;
            case '3': return 3;
            case '4': return 4;
            case '5': return 5;
            case '6': return 6;
            case '7': return 7;
            case '8': return 8;
            case '9': return 9;
            case 'A': return 10;
            case 'B': return 11;
            case 'C': return 12;
            case 'D': return 13;
            case 'E': return 14;
            case 'F': return 15;
      }
      return 0;
}

int hexToDecimal(char hex[], size_t size) {
      // complete the function body

      return 0;
}
```