Remember to bring this along to your lab. Prepare your report before attending the lab!

CS2100: Computer Organisation

Lab #2: Debugging using GDB II

(Week 5: 9 – 13 Sep 2024)

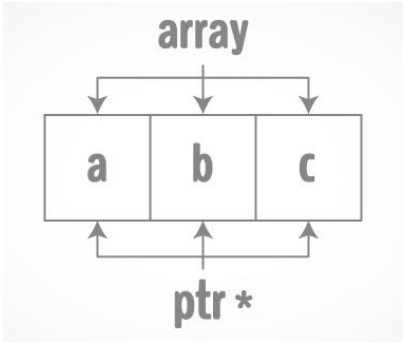
[This document is available on Canvas and course website https://www.comp.nus.edu.sg/~cs2100]

Name: _____ Student No.: _____

Lab Group: _____

C Arrays

Arrays are data structures that store fixed-size sequential collections of elements of the same type. While an array simply stores a collection of data, it is often more useful to think of the collection as a collection of variables of the same type.

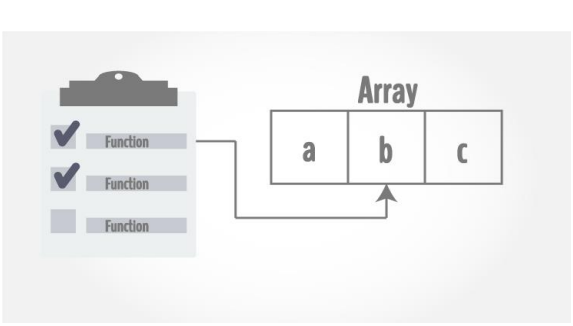


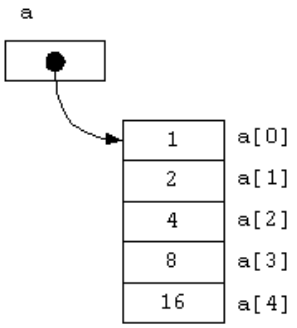
Instead of declaring individual variables, eg. `number0`, `number1`... `number99`, we can declare a single array variable `numbers` and use `numbers[0]`, `numbers[1]`,…`numbers[99]` to represent individual variables. A specific element in an array is accessed by an index which starts from 0.

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

C Functions and Arrays

In C programming, both a single array element or an entire array can be passed to a function. A single value will be passed by value, whereas a whole array is always passed as a reference (think pointer) to the first element of the array. In other words, the array itself is represented by a pointer 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. Download the files **lab2a.c, lab2b.c and lab2c.c** from Canvas.

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

3. What is the output of the program?

```
4
```

4. Which line in the code should you change to get output **"2"** instead? Show the changed line.
   Note: The output should be related to the **`ageArray`**. Do not hardcode "2" in your code!

```
display(ageArray[0]);
```

5. What is the purpose of the unary operator **`sizeof`**?
   What datatype will **`sizeof`** give the value **"1"** for all architectures?

```
compute the size of the operand
char
```

6. Can you get the number of elements in **`ageArray`**? Write a modified main function below to produce the following output. Show your lab TA the output of the code.

```
2
Size of the array is 4
```

   Note: The output **"2"** and size of array (i.e., **4**) are related to **`ageArray`**. Do not hardcode the value "2" and "4" in your code!

```
int main() {
 int ageArray[] = { 2, 15, 4, 5 };
 display(ageArray[0]);
 int sizeArray = sizeof(ageArray) / sizeof(ageArray[0]);
 printf("Size of the array is %d\n", sizeArray);
}
```

7. Compile **lab2b.c** with **`gcc`** using the following command: **`gcc -o lab2b lab2b.c`**

8. Give 2 ways of displaying the stored value of the first element of an array.
Give 2 ways of displaying the address value of the first element of an array.

```
printf(*a) or printf(a[0])
printf(a) or printf(&a[0])
```

9. Can you define the function **`hexToDecimal(char hex[], size_t size)`** in **lab2b.c**, using pointers 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.*

```
int hexToDecimal(char hex[], size_t size) {
 // complete the function body
 int decimal = 0;
 int base = 1; // 16^0 = 1
 char *ptr = hex + size - 1;

   while (ptr >= hex) {
      decimal += hexVal(*ptr) * base;
      base *= 16;
      ptr--;
   }
   return decimal;
}
```

10. 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?

```
function only has a pointer to the first element of the array, not the array itself
(doesn't have info of the original array)
need to explicitly pass in size of array because sizeof no longer works
```

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

```
%zu
```

12. Compile **lab2c.c** using the following command: **`gcc –o lab2c –DTEST0 lab2c.c.`** What does the option **`–DTEST0`** do? Hint: read the man page of gcc, i.e. issue the command: **`man gcc`**.

compile the section of code under #if defined TEST0

13. Execute **`lab2c`** and report what happened. Explain how the output was obtained.

Rhis is a test!

modify the first character of the string to be R

14. Now recompile **lab2c.c** with: **`gcc –o lab2c –DTEST1 lab2c.c`**. Execute **`lab2c`** and report what happened. Explain how the output was obtained.

Segmentation fault
str_as_pointer points to a string literal which cannot be modified
cannot write to a memory that is read only

15. Now recompile **lab2c.c** with: **`gcc –o lab2c –DTEST2 lab2c.c`**. Execute **`lab2c`** and report what happened. Explain how the output was obtained.

his is a test!
increment the pointer to point to the next character in the array
printf prints starting from the current position of the pointer

16. Now recompile **lab2c.c** with: **`gcc –o lab2c –DTEST3 lab2c.c`**. Report what happened. Explain why.

error: lvalue required as increment operand
str_as_array is a character array not a pointer, cannot increment the array to point to a different location in memory

Marking Scheme: Report – 16 marks; correct output – 4 marks; Total: 20 marks.