

CSC3320 System Level Programming

Lab Assignment 9 - Post-Lab

Kevin Gallardo

Due at 11:59 pm on Sunday, March 21, 2021

Purpose: Learn how to use array in C. Understand the basic memory address in C.

Part 1:

Write a C program named as `getMostFreqChar.c` that finds the most frequent letter from the input via ignoring the case sensitive and prints out its frequency. For example, sample outputs could be like below

```
$cat test.txt
```

This is a list of courses.

CSC 1010 - COMPUTERS & APPLICATIONS

```
$/getMostFreqChar test.txt
```

The most frequent letter is 's'. It appeared 8 times. Run the C program,

attach a screenshot of the output in the answer sheet.

```
+
+  GSD Computer Science
+  Instructional Server
+  SNOWBALL.cs.gsu.edu
+
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ vi getMostFreqChar.c
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ vi getMostFreqChar.c
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ vi getMostFreqChar.c
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ cat test.txt
this is a tasty test
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ vi test.txt
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ ./getMostFreqChar test.txt
-bash: ./getMostFreqChar: No such file or directory
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ ./getMostFreqChar test.txt
-bash: ./getMostFreqChar: No such file or directory
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ vi getMostFreqChar.c
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ cc getMostFreqChar.c
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ ls
a.out  cac3320  debug  files.tar  fn.txt  getMostFreqChar.c  homeworks  Lab3  Lab4  mandatabase.txt  midterm  program  public  q1  q1.c  splitTime  splitTime.c  test  test.txt  txtfiles  txtfiles.tar.gz
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$ ./a.out test.txt
This is a list of courses.
CSC 1010 - COMPUTERS & APPLICATIONS
The Most frequent letter is 's'. It appeared 8 times.
(kgallardowepsteri@gsuad.gsu.edu$ snowball -j$
```

Part 2:

When a variable is stored in memory, it is associated with an address. To obtain the address of a variable, the & operator can be used. For example, &a gets the memory address of variable a. Let's try some examples.

Write a C program addressOfScalar.c by inserting the code below in the main function.

Questions:

1) Run the C program, attach a screenshot of the output in the answer

sheet.

```
+
|  GSU Computer Science
|  Instructional Server
|  SNOWBALL.cs.gsu.edu
+
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ vi addressOfScalar.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cc addressOfScalar.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ls
addressOfScalar.c  debug      getMostFreqChar  Lab3      midterm  ql      splitTime.c  txtfiles
a.out             file.txt  getMostFreqChar.c  Lab4      program  ql.c    test        txtfiles.tar.gz
csc3320           fn.txt   homeworks          mandatabase.txt  public   splitTime  test.txt
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./a.out
address of charvar = 0x7ffc09d303af
address of charvar - 1 = 0x7ffc09d303ae
address of charvar + 1 = 0x7ffc09d303b0
address of intvar = 0x7ffc09d303a8
address of intvar - 1 = 0x7ffc09d303a4
address of intvar + 1 = 0x7ffc09d303ac
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

2) Attach the source code in the answer sheet

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

```
int main() {
```

```
    char charvar = 'a';
```

```
    printf("address of charvar = %p\n", (void *)&charvar);
```

```
    printf("address of charvar - 1 = %p\n", (void *)&charvar - 1);
```

```
    printf("address of charvar + 1 = %p\n", (void *)&charvar + 1);
```

```
    int intvar = 1;
```

```
    printf("address of intvar = %p\n", (void *)&intvar);
```

```
    printf("address of intvar - 1 = %p\n", (void *)&intvar - 1);
```

```
    printf("address of intvar + 1 = %p\n", (void *)&intvar + 1);
```

```
    return 0;
```

```
}
```

2) Then explain why the address after intvar is incremented by 4 bytes instead of 1 byte.

The address after intvar is incremented by 4 bytes because the size of an integer is 4 bytes.

```
1 // initialize a char variable, print its address and the next address
2 char charvar = '\0';
3 printf("address of charvar = %p\n", (void *)&charvar);
4 printf("address of charvar - 1 = %p\n", (void *)&charvar - 1);
5 printf("address of charvar + 1 = %p\n", (void *)&charvar + 1);
6
7 // initialize an int variable, print its address and the next address
8 int intvar = 1;
9 printf("address of intvar = %p\n", (void *)&intvar);
10 printf("address of intvar - 1 = %p\n", (void *)&intvar - 1);
11 printf("address of intvar + 1 = %p\n", (void *)&intvar + 1);
12
```

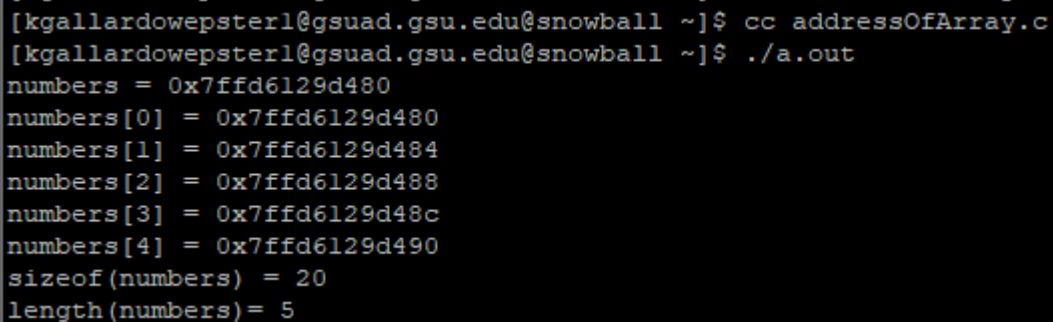
Part 3:

Write a C program addressOfArray.c by inserting the code below in the main function.

```
1 // initialize an array of ints
2 int numbers[5] = {1,2,3,4,5};
3 int i = 0;
4
5 // print the address of the array variable
6 printf("numbers = %p\n", numbers);
7
8 // print addresses of each array index
9 do {
10     printf("numbers[%u] = %p\n", i, (void *)&numbers[i]);
11     i++;
12 } while(i < 5);
13
14 // print the size of the array
15 printf("sizeof(numbers) = %lu\n", sizeof(numbers));
```

Questions:

1) Run the C program, attach a screenshot of the output in the answer sheet.



```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ cc addressOfArray.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./a.out
numbers = 0x7ffd6129d480
numbers[0] = 0x7ffd6129d480
numbers[1] = 0x7ffd6129d484
numbers[2] = 0x7ffd6129d488
numbers[3] = 0x7ffd6129d48c
numbers[4] = 0x7ffd6129d490
sizeof(numbers) = 20
length(numbers)= 5
```

2) Check the address of the array and the address of the first element in the array.
Are they the same?

Address of array : 0x7fff11ae4f20

Address of the first element in the array: 0x7fff11ae4f20

Yes, both addresses are the same.

3) Write down the statement to print out the length of the array by using sizeof operator.

```
printf("length(numbers)= %lu\n", sizeof(numbers)/sizeof(numbers[0]));
```

Submission:

- ✂ Upload an electronic copy (pdf) of your answer sheet to the folder named “Lab 9” in Google Classroom
- ✂ Please add the lab assignment number and your name at the top of your answer sheet.
- ✂ Upload the C files getMostFreqChar.c, addressOfArray.c and addressOfScalar.c to the folder named named “Lab 9” in Google Classroom
- ✂ Name your file in the format of Lab9_ **FirstnameLastname** (e.g Lab9_FilRondel.pdf)