CSC3320 System Level Programming

Lab Assignment 9 - Post-Lab

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

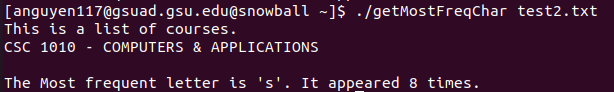
$cat test.txt

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.



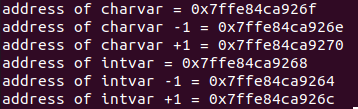
Part 2:

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.

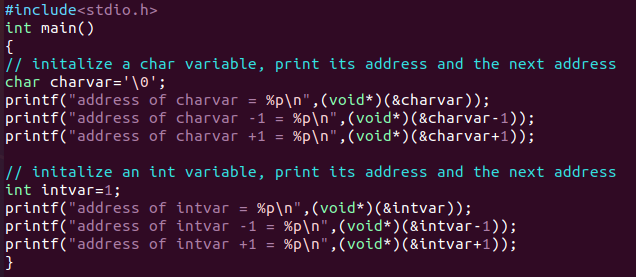
Write a C program addressOfScalar.c by inserting the code below in the main function.

Questions:

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



2) Attach the source code in the answer sheet

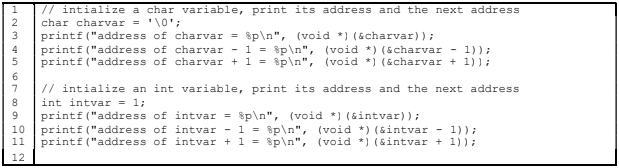


3) Then explain why the address after intvar is incremented by 4 bytes instead of 1 byte.

**Character variables takes 1 byte of memory so charvar increment by 1 byte.**

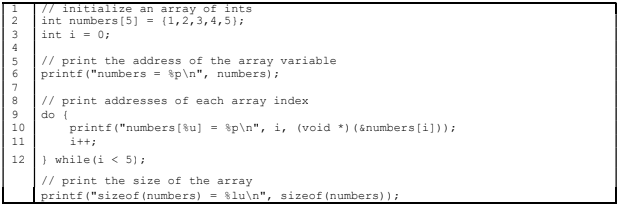
**Integer takes 4 bytes memory so intvar increments by 4 bytes instead of 1.**

**Size of int datatype is 4 bytes so when we increment the address by 1, it will increment 4 bytes.**



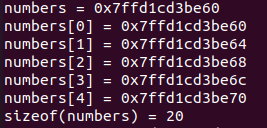
Part 3:

Write a C program addressOfArray.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.



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

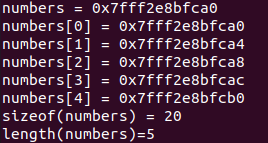
**Since the address of the array is the address of its first element are identical, they are the same.**

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]));**

**Dividing “size of array” by “size of individual array elements” gives “length of an array”.**

**length(numbers)= 5**



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)