

Aleeza Iftikhar
Lab Assignment 9 - Post-Lab
CSC3320 System Level Programming

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.

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$ ./getMostFreqChar
This is a list of courses.
CSC 1010 - COMPUTERS & APPLICATIONS

MAX counted letter is 's' and total count is '8'[aiftikhar2@gsuad.gsu.edu@snowba
ll Lab9]$
```

Part 2:

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.

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$ gcc -o addressOfScalar -g addressOfSc
lar.c
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$ ./addressOfScalar
Hello World
address of charvar = 0x7ffc6094407f
address of charvar - 1 = 0x7ffc6094407e
address of charvar + 1 = 0x7ffc60944080
address of intvar = 0x7ffc60944078
address of intvar - 1 = 0x7ffc60944074
address of intvar + 1 = 0x7ffc6094407c
```

2) Attach the source code in the answer sheet

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$ cat -n addressOfScalar.c
 1  #include <stdio.h>
 2
 3  int main(void) {
 4      printf("Hello World\n");
 5      // initialize a char variable, print its address and the next address
 6      char charvar = '\0';
 7      printf("address of charvar = %p\n", (void *)(&charvar));
 8      printf("address of charvar - 1 = %p\n", (void *)(&charvar - 1));
 9      printf("address of charvar + 1 = %p\n", (void *)(&charvar + 1));
10
11      // initialize an int variable, print its address and the next address
12      int intvar = 1;
13      printf("address of intvar = %p\n", (void *)(&intvar));
14      printf("address of intvar - 1 = %p\n", (void *)(&intvar - 1));
15      printf("address of intvar + 1 = %p\n", (void *)(&intvar + 1));
16      return 0;
17  }
```

3) Then explain why the address after intvar is incremented by 4 bytes instead of 1 byte.
Because character occupies space of 1 byte, whereas integer occupies 4 bytes space.

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.

```
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$ gcc -o addressOfArray -g addressOfArray.c
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$ ./addressOfArray
Hello World
numbers = 0x7fff0a5ae7f0
numbers[0] = 0x7fff0a5ae7f0
numbers[1] = 0x7fff0a5ae7f4
numbers[2] = 0x7fff0a5ae7f8
numbers[3] = 0x7fff0a5ae7fc
numbers[4] = 0x7fff0a5ae800
sizeof(numbers) = 20

Length of array: 5
[aiftikhar2@gsuad.gsu.edu@snowball Lab9]$
```

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

Yes, because address of array is starting address of array which is same as address of first element of array.

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

It will be `sizeof(array)` divided by size of first element of array, which in this case will be:
`sizeof(numbers)/sizeof(numbers[0])`