CSC3320 System Level Programming
Lab Assignment 9 - Post-Lab

Name: Venkata Mani Mohana Rishitha Srikakulapu
Lab # 9 – Out of Lab Assignment Report

Part 1:

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

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi getMostFreqChar.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ gcc getMostFreqChar.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out test.txt
The most frequent letter is 's' . It appeared 8 times.
```

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

int main(int argc, char *argv[] ){
        char const* const txtFile = argv[1];
        FILE* file = fopen(txtFile, "r");

        int array[26] = {0},l;

        while(1){
        l = fgetc(file);
        if(l==EOF)
                break;

        if(l>='a'&&l<='z')
        array[l-'a']++;
        else if(l>='A'&& l<='Z')
        array[l-'A']++;
        }

int i, max = array[0];
char j;
        for(i=0;i<26;i++){
                if(array[i]>max){
                        max = array[i];
                        j = i+'a';
                }
        }

printf("The most frequent letter is \'%c\' . It appeared %d times.\n",j,max);

fclose(file);

return 0;
}
```

Part 2:

Questions:

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

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi addressOfScalar.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ gcc addressOfScalar.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
address of charvar = 0x7ffcd4e6047f
address of charvar -1 = 0x7ffcd4e6047e
address of charvar + 1 = 0x7ffcd4e60480
address of intvar = 0x7ffcd4e60478
address of intvar - 1 = 0x7ffcd4e60474
address of intvar + 1 = 0x7ffcd4e6047c
```

2) Attach the source code in the answer sheet

```
#include<stdio.h>
int main(){

// initialize a char variable, print its address and the next
address
char charvar = '\0';
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));

// initialize an int variable, print its address and the next
address
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;
}
```

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

Int variable is of type int and occupies 4 bytes of memory whereas char variable is of type char and occupies 1 byte in memory. So, when intvar is incremented by 1 , its address gets incremented by 4 bytes.

Part 3:

Questions:

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

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi addressOfArray.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ gcc addressOfArray.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
numbers = 0x7ffe7a41cb30
numbers[0] = 0x7ffe7a41cb30
numbers[1] = 0x7ffe7a41cb34
numbers[2] = 0x7ffe7a41cb38
numbers[3] = 0x7ffe7a41cb3c
numbers[4] = 0x7ffe7a41cb40
sizeof(numbers) = 20
```

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

Yes, the address of the array and address of the first element of the array are same i.e. 0x7ffe7a41cb30.

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

```
printf("length of the array is = %1u\n",
sizeof(numbers)/ sizeof(numbers[0]));
```

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi addressOfArray.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ gcc addressOfArray.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
numbers = 0x7ffd43c1f1a0
numbers[0] = 0x7ffd43c1f1a0
numbers[1] = 0x7ffd43c1f1a4
numbers[2] = 0x7ffd43c1f1a8
numbers[3] = 0x7ffd43c1f1ac
numbers[4] = 0x7ffd43c1f1b0
sizeof(numbers) = 20
length of the array is = 5
```