Lab 3: Introduction to Data Types and I/O

1. **Write a C program (*sizeDataTypes.c*) that prints the size of *short*, *int*, *char*, *double*, *long* and *long double* using *sizeof* operator and show that the following guidelines are true.** **4 pts**

   sizeof(short) < sizeof(int)
   sizeof(char) < sizeof(short)
   sizeof(long) > sizeof(int)
   sizeof(long double) > sizeof(double)

2. **What will be final values of *'a'*, *'b'*, *'c'* and *'size'* in the following code snippet?**

   int a =3; // **finalValue.c** **4 pts**
   int b =4;
   double c = ++a + b++;
   int size = sizeof(c/a);

3. **Consider the following program that asks the user for 10 positive integers (grade). Modify the program as follows:**
   a. The program will be able to read any number of inputs from the keyboard. -1000 will indicate the end of processing. **2 pts**

```
#include <stdio.h>        // AverageGrade.c

int main(){
   int i = 0;
   int grade;
   printf("Please input 10 grades :\n");
   while(i < 10) {
      scanf("%d", &grade);
      i ++;
   }
   return 0;
}
```

   b. As the user keeps on typing integers, the program will keep on summing and averaging the numbers typed so far. For example, in the following screenshot, the user typed 10, so, the sum and average are both 10. Next, the user typed 20, the sum and average were calculated considering the previous input '10'. This will continue till the user types -1000. .You need to find the sum and average in double.

   4 pts

```
syasmin@cscd-linux01:~/CPractice$ ./AverageGrade
Please input a number of grades:
10
Sum is 10.000000
Average is 10.000000
20
Sum is 30.000000
Average is 15.000000
45
Sum is 75.000000
Average is 25.000000
56
Sum is 131.000000
Average is 32.750000
-1000
syasmin@cscd-linux01:~/CPractice$
```

4. **What is the problem with the following C program? How will you fix it? 4 pts(2+2)**

```
#include <stdio.h> //charBug.c
int main( )
{
   int c;
   char dummy[10];

   printf( "Enter a value :");
   c = getchar( );

   while( c != 'q' && c != 'Q' ) {
          printf( "You entered: ");
          putchar( c );
          printf("\n");
          printf("Enter a value :");
          c = getchar();
   }
   return 0;
}
```

5. **What are the differences between 'gets' and 'fgets' functions. Rewrite the following code with gets. 4 pts (2+2)**

```
#include <stdio.h>    //getsFgets.c

int main() {
      char name[100];
      printf("Please enter a name:");
      fgets( name, 100, stdin);
      printf("The name you entered is %s", name);

return 0;

}
```

6. Write a C program that reads a sentence from the keyboard until a newline character is met (i.e., '**Enter**' button from the keyboard is pressed). Next, the user is asked to type any alphabet and the program will count the number of occurrences of that particular alphabet in the sentence. The program will ignore case in alphabets (i.e., uppercase or lowercase). For example, if the user types "Hello World" and wants to look for the number of occurrences of 'O'/ 'o' in the sentence, the program will generate the following output.

8 pts

```
syasmin@cscd-linux01:~/CSCD240$ ./countChar
Type a sentence:
Hello World!!!
Type a character that you'd like to find the number of occurrences in the sentence:
o
Alphabet o has a frequency of 2
syasmin@cscd-linux01:~/CSCD204$
```

Consider the following C code **countChar.c**. The comments in code show the steps you need to write to make the code work.

```
#include <stdio.h>


#define MAXSIZE 100

int main(){
    char ch;
    int i;
    char sentence[MAXSIZE];
    int alphabetsCount = 0;

    printf("Type a sentence: \n");

    // write the code that reads a sentence from the keyboard 2pts

    printf("Type a character that you'd like to find the number of occurrences
in a sentence:\n");

    // write the code that reads a character from the keyboard  2 pts

    // write the code that counts the number of character in the sentence; 2 pts

    // ignore uppercase or lowercase  2 pts

    printf("Alphabet  %c has a frequency  %d\n", ch, alphabetsCount);

    return 0;

}
```

Submission:

- All answers (whenever you need to write codes) should be submitted as complete C codes named as **sizeDataTypes.c (Question 1)**, **finalValues.c (Question 2)**

AverageGrade.c (Question 3), charBug.c (Question4), **getsFgets.c (Question 5)** and **countChar.c (Question6)** respectively.

- **Part of Question 5** should be submitted in a PDF file named **Lab3.pdf**.
- Full submission should contain all C files and pdf file zipped as follows: your last name, first letter of your first name, Lab3.pdf (i.e., YasminSLab3.pdf). You should turn in through the EWU Canvas system.
- Submission deadline is **Tuesday, January 6, 2018**. **No late submission will be accepted**.