EE2028A

C Programming

Laboratory Exercise (LAB-III)

Name: Anurag Chatterjee

Matriculation Number: A0200533Y

Submission instructions:

1. **Test your code on your computer first** before submitting.
2. You must name your functions exactly as the question states.
3. **DEADLINE**: **Tuesday18 Feb 2020 / Thursday 20 Feb 2020**
4. **LumiNUS "Lab 3 Assignment Submission Folder"**
5. **Grading: Your assignment will be graded out of 50 marks and the final weight of this assignment is 20%.**
6. You are expected to follow the guidelines given below:
    a. Use meaningful variable names while programming. It's a good practice to develop good programming skills and enables readability.
    b. Explain the code with proper comments; Comments must be meaningful and descriptive;
    c. Please adhere to the report deadlines and any late submissions are not accepted.
7. Please prepare the report in **PDF**format **only**.
8. Submit the following:

Submit the compressed file	Contains
MATRICULATION_NUMBER_ASSIGNMENT 3_NAME (First Name).zip	REPORT_MATRICULATION_NUMBER_ASSIGNMENT3_NAME (First Name).**pdf**
	Your working C code, **ONLY .c file**

What you need to add into this report for submission? - YOUR OUTPUT:

a. Program Code (attach in the ZIP file – see the guidelines for submission) addition to the .c file that you need to submit
   - Code should be well written with meaningful variables and comments.

b. In THIS report, **screenshot your results and paste.** Make sure it is visible, readable and clear.
   - **For Question1**Your code and your screenshots for 3 random cases, screenshots should be for the entire code, not for sub problems.

DO NOT FORGET TO SEND YOUR .C FILE WITH ALL YOUR WORKING CODES BESIDES ATTACHING THE CODES INTO THE REPORT.

NOTE: Start your answers from here. Use as much space as needed.

PROBLEM 1:

```c
#include<stdio.h>
#include<stdlib.h>
#include<time.h> //for random number generation
void Print_Arr(int Signal[50], int size) {
    int i;
    for (i = 0;i < size;i++) {
        printf("%d\t",Signal[i]); //Creates new tabs whenever the loop iterates
        i+=1; //i value increments by 1
        if (i % 10 == 0) {
            printf("%d\n"); //If i is divisible by 10 then a new line should be created else another tab
        }
        else {
            printf("%d \t");
        }
    }
    printf("\n");
}
void Max_Min(int Signal[50],int size) {
    int i;
    int maxvalue,minvalue;
    maxvalue=0;
    minvalue=0;
    for (i=0;i<size;i++) {
        if (Signal[i] > maxvalue) {
        maxvalue = Signal[i];
        }
        if (Signal[i] < minvalue) {
        minvalue = Signal[i];
        }
    }
    printf("Max Signal Value : %d\n",maxvalue);
    printf("Min Signal Value : %d\n",minvalue);
}
int Zero_Bias(int Signal[50],int size) {
    int storingarray[50]; //stores the result array for zeros
    int i;
    int negval[50]; //array storing negative value elements
    int count=0; //number of negative values
```

```
    for (i=0;i<size;i++) {
        if ((Signal[i]) >= 0) {
            storingarray[i] = Signal[i]; // if signal value is 0 make it 0 else no
        }
        else {
        storingarray[i] = 0;
        }
        if (Signal[i] < 0) {
            storingarray[i] = 0;
            count+=1;
            negval[count]=i;//increases negative count for elements below 0
        }
    }
    printf("Negative at indices: \n");
    Print_Arr(negval, count);
    printf("Total number of negative values: %d \n", count);
    Print_Arr(storingarray, 50);
}
int Threshold_Detect_Count(int Signal[50],int size) {
    int exceeded4=0;
    int i;
    int belowminus4=0;
    for (i=0;i<size;i++) {
        if (Signal[i]< -4) {
        belowminus4+=1;
        }
        else if (Signal[i] > 4) {
        exceeded4+=1;
        }
    }
    if ((exceeded4 >= 10) && (belowminus4 >= 10)) {
        printf("Original signal category: HPHA");
        printf("\n");
    }
    else if ((exceeded4 <= 9) && (belowminus4 <= 9)) {
        printf("Original signal category: HPLA");
        printf("\n");
        printf("Original signal category: Normal");
        printf("\n");
```

```
    }
}
int Missing_Samples(int Signal[50],int arrsize) {
    int i,x;
    int missing[50];
    int missingnumbers = 0;
    for (x=-10;x<=10;x++) {     //finds numbes between -10 and 10
        for (int i = 0; i < arrsize; i++) {     //iterates through Signal array
            if (Signal[i] == x) {                   //if number is found, break the loop
                break;
            }
            else if (i == (arrsize - 1)) {  //if the number is found at the end of Signal
                if (Signal[i] != x) {
                    missingnumbers+=1;
                    missing[i] = x;
                    break;
                }
            }
        }
    }
    printf("Num.of samples missing in the range[-10, 10]: %d \n", missingnumbers);
    Print_Arr(missing, missingnumbers);
}
int main() {
        /* Intializes random number generator */
        srand(time(NULL));

        int Signal[50];

        for (int i = 0; i < 50; i++) {
            Signal[i] = rand() % 21 - 10;           //each element is a random number from range -10 to 10 (21 numbers inlcuding 0)
        }
    printf("Signal Analysis\n");
    printf("Original Signal\n");
        Print_Arr(Signal, 50);
```

```
        for (int i = 0; i < 50; i++) {
            Signal[i] = rand() % 21 - 10;          //each element is a random number from range -10 to 10 (21 numbers inlcuding 0)
        }
    printf("Signal Analysis\n");
    printf("Original Signal\n");
        Print_Arr(Signal, 50);


        Max_Min(Signal, 50);


      Zero_Bias(Signal, 50);


        Threshold_Detect_Count(Signal, 50);


        Missing_Samples(Signal, 50);


        return 0;
}
```

```
Signal Analysis
Original Signal
4   10  -6  10  -4  10  -8  10  -8  10  -7  10  5   10  0   10  3 1
0   -5  10  -10 10  10  10  -1  10  8   10  -2  10  -1  10  0   101
0   -8  10  10  10  0   10  -8  10  1   10  8   10  -3  10
Max Signal Value : 10
Min Signal Value : -10
Negative at indices:
0   10  4   10  7   10  10  10  20  10  23  10  28  10  31  10  371
0   41  10  43  10  48  10
Total number of negative values: 23
4   10  0   10  0   10  0   10  0   10  0   10  5   10  0   10  3 1
0   0   10  0   10  10  10  0   10  8   10  0   10  0   10  0   101
0   0   10  10  10  0   10  0   10  1   10  8   10  0   10
Original signal category: HPHA
Num.of samples missing in the range[-10, 10]: 0
```