

CSE 1310 INLAB 09

Make sure to save source code for each checkpoint into a separate file. The naming convention of the files should be 'functions-chk1.c', 'functionschk2.c', etc. For example, after you have completed a checkpoint1, then copy 'functions-chk1.c' to 'functionschk2.c' using the Unix copy command.

Open a session on 'omega' and using the 'vi' editor, create a new file by name 'functionschk1.c'.

Write a main() function that calls another function, menu(). Remember to include the libraries that are needed. Write one more function menu(), which should prompt the user for a choice from 1 to 4. Choice 1 corresponds to calculating average, choice 2 is used to generate random numbers, choice 3 is used for finding the square root, and choice 4 is used to exit the program. (See the sample output at the end of this lab). The sample output given does not indicate that you should get the same random values for the temperatures. Outline of the program is given below:

```
/******
```

Filename: functions.c

Description: This program makes use of the concepts of functions to solve problems in C programming. It involves writing functions for specific operations namely calculating the average of a set of temperature values, generating a specific number of temperature values, and calculating the square root.

Programmer:

Lab Instructor:

Lab Section:

```
*****/
```

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

```
void menu (void); /* function prototype */
```

```
int main(void)
```

```
{
```

```
    menu();
```

```
    return (0);
```

```
}
```

```
void menu(void)
```

```
{
```

```
    int choice = 1;
```

```
    while ( )
```

```
    /* Note: fill the correct condition in the while loop to ensure that only cases 1 to 4 are selected and anything else should stop the loop execution */
```

```
    {
```

```
        printf("Choose one of the following\n");
```

```
        printf("1. Average\n");
```

```
        printf("2. Random\n");
```

```
        printf("3. Square root\n");
```

```
        printf("4. Exit\n");
```

```
        scanf("%d", &choice);
```

```
        switch (choice )
```

```

        {
            case 1:
                /* average() will be called here */
                break;
            case 2:
                /* Random() will be called here */
                break;
            case 3:
                /* squareRoot() will be called here */
                break;
            case 4:
                exit(0);
            default:
                printf("Wrong choice. Exiting the program.\n");
        } /* switch */
    } /* while */
} /* menu */

```

After you type the above code, you need to make some more changes. There is a **Note** in above code. Read this **Note** carefully, do whatever it says and then remove it. Look at the previous lab if you need some help for doing this.

Complete the program and compile it. Once you ensure that there are no errors, signal the grader.

Checkpoint 1

Once you finish the 'switch' statement, define three functions 'average', 'Random' and 'squareRoot' outside the main function and write a stub for each of the functions, i.e., at this point they do not perform any operations. They will have only a printf() statement that displays a message saying that the user is in that particular function. Now, when you compile and execute your program 'functions-chk2.c', it should prompt the user to enter a choice and when a choice is selected, it should call the corresponding function in menu() in order to display the message for the corresponding function.

Hint: Use following help for writing the average() function—

At the beginning of the file, just below the prototype for menu() function, write the prototype for average() function as follows:

```
void average(void);
```

After the end of menu() function, write the definition of average() function as follows:

```

void average(void)
{
    printf("This function prints the average.\n");
}

```

You will do the same procedure for other two functions. When you have the output with each stub working, signal the grader.

Checkpoint 2

For this checkpoint, you will be working on the average function. This function will prompt the user for 10 temperature values, find the average and print the results to the screen using two decimal places to the right. Remove the printf() statement used here initially and replace it with code to calculate the average, given the 10 temperature values as input. You can use a 'for' loop and have a variable inside the 'for' loop to read in the temperature values. When your average function is working, signal the grader.

Checkpoint 3

Now you'll work on the Random function. Here, remove the printf() statement that you used initially and include the code for the Random function as explained below to generate 6 random temperature values . Your function definition will prompt and accept the number of random values to be generated at the command prompt (in this case, 6 values). Make sure to include the 'stdlib.h' header file which contains the definition of the rand() function. After you have the Random number function working, display its results and signal the grader.

```
void Random(void)
{
    int i, num, random_num;
    printf("Enter the number of random values to be generated:\n");
    scanf("%d", &num);
    for(i = 0; i < num; i++)
    {
        random_num = rand() % 100;
        /* use a printf statement to print the value of random_num and
        remove this comment */
    }
}
```

Checkpoint 4

For the final checkpoint, you can use a 'for' loop to get 6 temperature values from the user. Determine the square root of sum of all these temperature values. Here is the outline of this function:

1. Use a 'for' loop to read the six temperature values.
2. Within this loop, add the value you read to another variable called sum.
3. After the loop, find square root of sum by using the function sqrt().
4. Make sure you include the 'math.h' header file to use function sqrt().
5. When you compile the program, use **gcc -lm functions-chk5.c** to compile the program. The option -lm is required to link the math library.

Make sure you declare all the variables you will be using at the beginning of this function. When you have a working program, signal the grader.

Checkpoint 5

Sample Output

Choose one of the following

1. Average
2. Random
3. Square root
4. Exit

1

Enter the temperature values: 100 15 25 76 100 60 70 56 65 77

The average is 64.4

Choose one of the following

1. Average
2. Random
3. Square root
4. Exit

2

The random temperatures are

65 32 45 65 55 76

Choose one of the following

1. Average
2. Random
3. Square root
4. Exit

3

Enter the temperature values: 65 32 45 65 55 76

The square root for the 6 temperature values is 18.38

Please choose one of the following

1. Average
2. Random
3. Square root
4. Exit

4

Exiting the program.