

**LAB 4**

**STRUCTURED PROGRAM DEVELOPMENT – COMPUTER-ASSISTED INSTRUCTION**

Bachelor Programme in Computer Science

Ebeten Alaga

2021-05-10

**A lab report in course DT131C Programming in C**

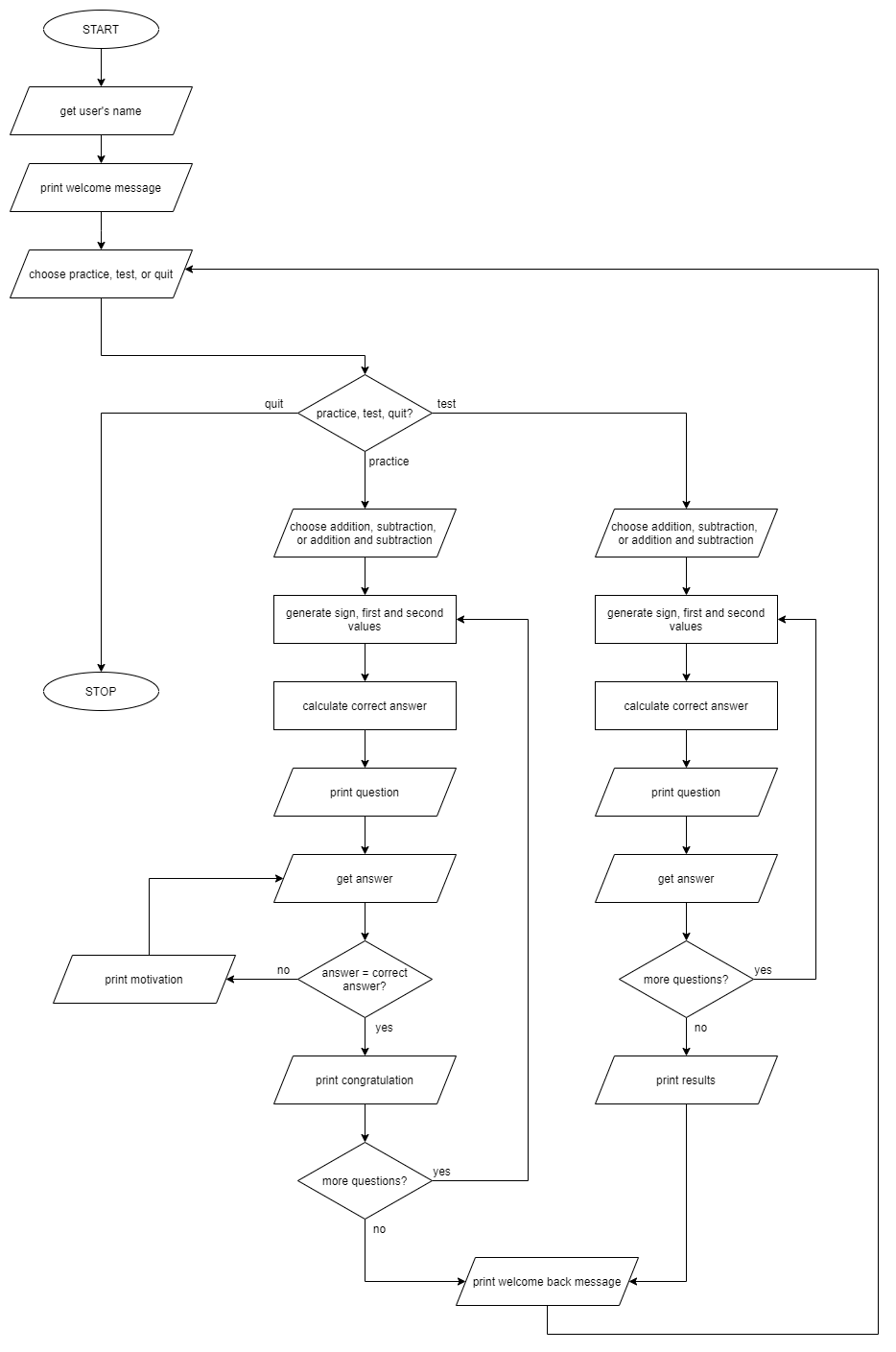
# 1. Introduction

This lab is to provide further practice on algorithm development, implementation and program testing by solving a more complicated problem.

The objectives of this lab include applying the top-down design to develop a solution to a complicated problem, implementation solutions to sub-problems in functions, and using arrays and strings.

# 2. Design

* Get the user’s name and welcome the user.
* Decide if it is a practice or test session. There is also an option to quit the program.
* Decide if it is addition, subtraction, or addition and subtraction.
* Generate the sign and values for the question.
* Calculate the correct answer.
* Print the question.
* Get answer from the user.
* If practice, print a message and get another answer if necessary.
* If there are more questions, begin from step 4 again. If there are no more questions, show test results.
* Welcome the user back.
* Begin from step 2 again.



# 3. Implementation

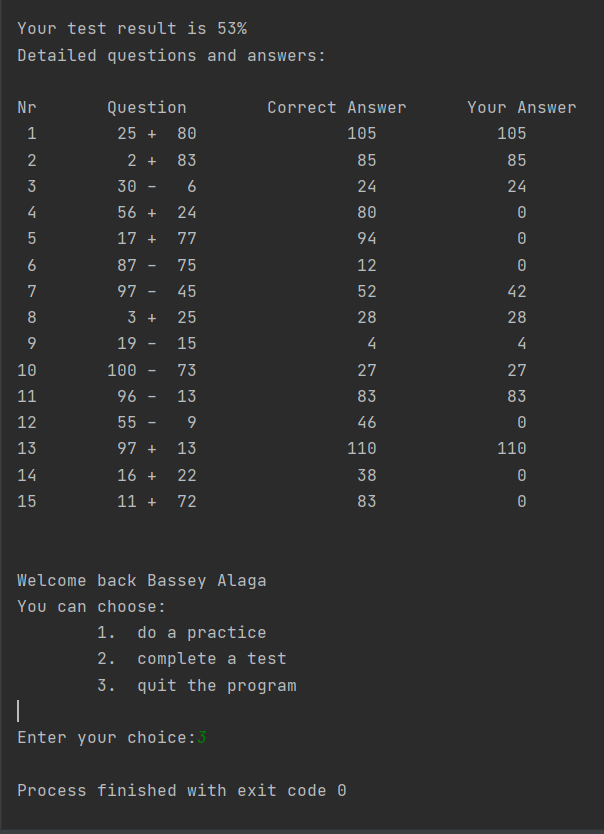
For this lab, I broke all the aspects of the problem into functions. Each of these functions performs a specific task that takes us a step closer to solving the problem.

The major parts of the program are the welcome area, which find out details about the user and how they intend to use the program.

The next part administers questions to the use. Note that the number of questions and other functionality of the program depend on the user’s input in the welcome area.

In solving these two major parts of the problem, they are divided into even smaller problems – example the welcome area is made up of different functions that get the user’s name and welcome them; and other functions that enquire about how the user intends to use the program. While asking questions, there are other functions that perform specific tasks such as setting the sign in the question or generating random numbers to use in the questions. There are also other functions to do tasks such as printing the results after a test or responses after questions in a practice.

# 4. Test and evaluation



Sample output showing the results printed after a test.

# 5. Conclusion and what I have learnt

In this lab, I improved on the use of functions in programs. I also learnt it is a good decision to let program flow be controlled in the main function, so rather than having each function calling other functions, I had the functions return values to main, and used those values as parameters when calling other functions. There were exceptions, such as when a function needed to have random numbers, in which case the call to the random number generator was made from within that function and the number(s) generated returned directly to it.

Some of the difficulties faced include error handling and having a default case in the switch-case construct.

# 6. References

No references were used in this report.