

**LAB 3 Report**

**Algorithm Development (C)**

Bachelor Programme in Computer Science

Ebeten Alaga

2021-05-10

**A lab report in course DT130B Programming in C**

# 1. Introduction

This lab aims to provide practice on algorithm development and representation in both flowcharts and pseudo codes. It also covers algorithm implementation in C.

It is divided into three tasks: multiplication table, computation of ex, bitwise operations.

# 2. Design

**Task 1:**

Decide on the size of the multiplication table, then create a 2D array. The array should be

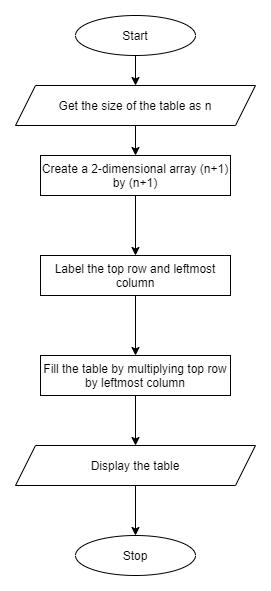
one size larger than the intended table to allow a row and column for displaying labels.

Next, label the top row and first column from one to the desired size of the table.

Fill the table by multiplying the labels and putting the result in the row / column

intersection.

Display the array.



**Task 2:**

Get the number, then display the result header.

Initialise count, sum, and top variables.

Now calculate the value of the numerator. Divide this value by the factorial of the count and add the result to the sum. Display the term and sum.

Repeat the last step while the top divided by the factorial is greater than 0.0000001.

Diagram

Description automatically generated

**Task 3:**

Read the status of the different systems from user inputs.

Perform bitwise operations on variable stat depending on the user inputs.

Convert stat from decimal to 8-bit binary and display it.

Compare the bit positions of stat and print the corresponding messages.

Diagram

Description automatically generated

# 3. Implementation

**Task 1:**

For this task, it is important to understand how arrays and nested loops work in C. We begin by creating an array to store the multiplication table and fill out the multiplication table use nested loops. The first loop fills out the rows, and the second loop fills out the columns.

We also format the displayed text using loops.

**Task 2:**

For task 2, we begin by initializing sum, top and count to 1. This is because we know the first time we calculate the exponent, we would be dividing the square root of 1 by one.

Subsequently we calculate the value of the numerator, divide it by the factorial and add the result to the sum.

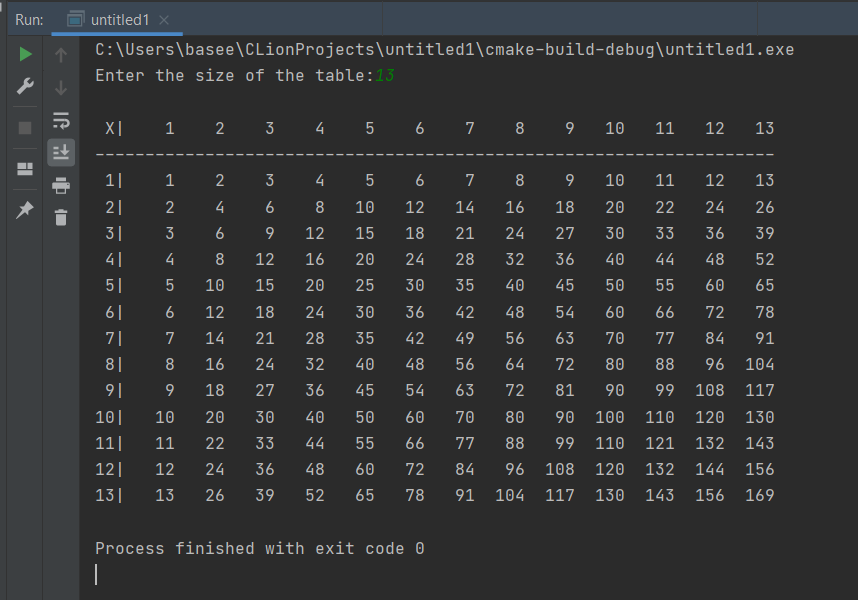
**Task 3:**

For task 3, we have a set of simple functions that perform simple tasks.

We have functions to perform bitwise operations for each indicator (e.g brake pads, outside temperature). We also have a function to read values from the keyboard, and functions to format and display the faults we discover.

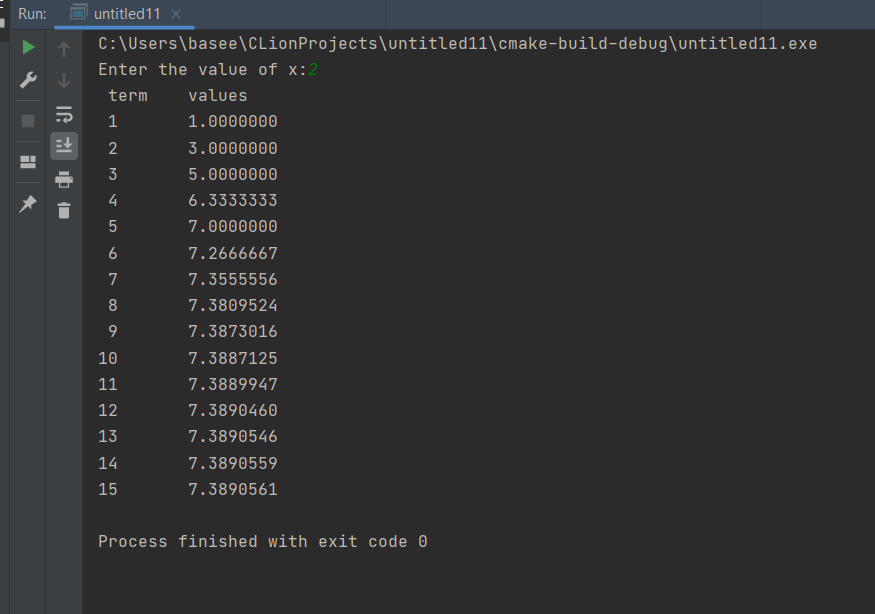
# 4. Test and evaluation

**Task 1:**



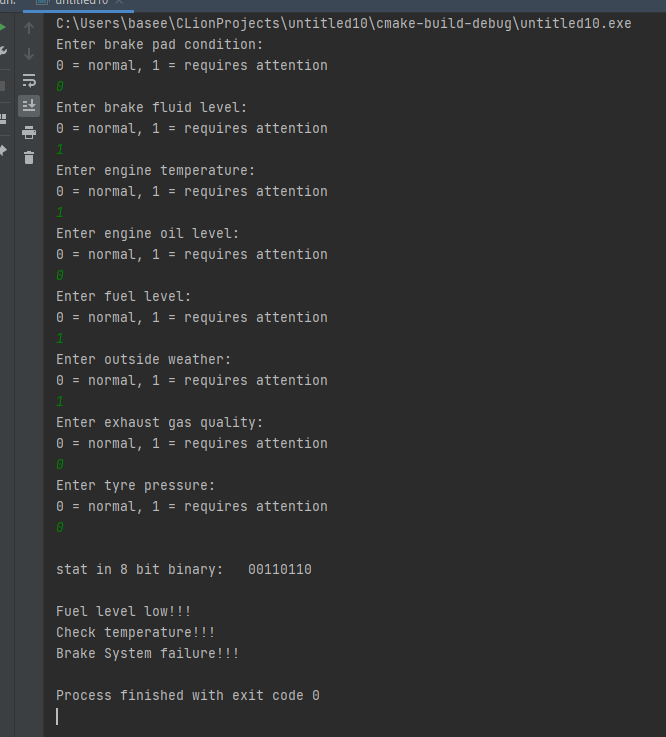
Sample output from task 1 using 13 as input.

**Task 2:**



Sample output from task 2 using 2 as input.

Task 3:



Sample output from task 3 with some inputs as normal and others as faulty.

# 5. Conclusion and what I have learnt

In completing this lab, I got to learn the bitwise operations in C, and by the end of the lab I began to appreciate the use of functions better.

While I have not fully grasped the use of header files, I can see the importance of using them and fully intend to study them in more detail going forward.

It is also clear that writing pseudocode that lists the steps needed to solve a problem simplifies implementation to a large extent, so that is one more thing to improve on.

# 6. References

No references were used for this report.