Lab 1

- Algorithm Development

Tomas Berggren

20191012

A lab report in the course DT555A Programming in C

1. Table of Contents

2. Introduction

This lab contains two tasks. The first one is a learning math tool. The second one prints multiplication tables.

For the basic grade (3) I have chosen Task 1 and Task 2.

---

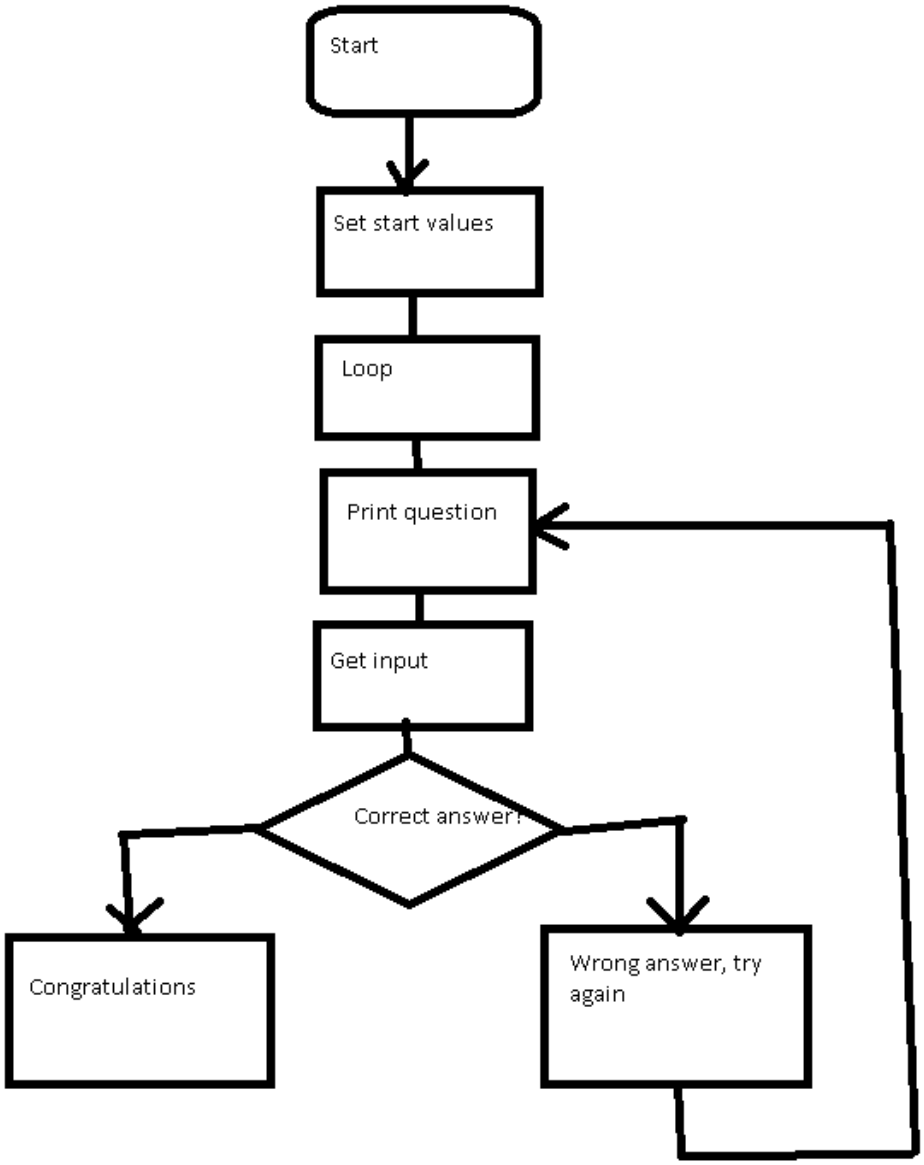
3. Design

3.1. Task 1

The aim with this program is to create a help tool for students learning math. The program should pick two random numbers, ask for the sum, control if it is correct or not and display either correct or wrong, try again. The algorithm is simple, set two numbers, display a questions and match the answer to the sum of the two numbers, if correct end, else new guess.

3.1.1. Pseduo code and flowchart

```
Use srand to get different random seed
Set rand_nr_a to random number between 0-100
Set rand_nr_b to random number between 0-100
Set stop to false
While stop is false
        Ask for the sum of rand_nr_a and rand_nr_b
        Input the answer
        If rand_nr_a + rand_nr_b == answer
                Print very good
                Set stop to true
         Else
                Print try again
endwhile
```



3.2. Task 2

This program should ask for input of how many multiplication tables should be listed. Those should then be listed. The algorithm here is also simple, input the number of tables, use this number in a loop as max condition, add one to the counter for each lap

3.2.1. Pseduo code and flowchart

Set stop to false

While stop is false

    Ask for input

    Input the answer and save into variable

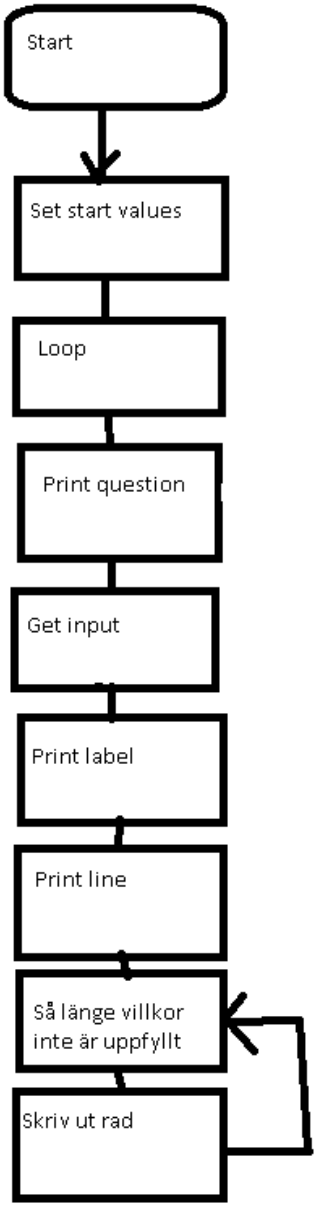
    Print first line as a label

    Print second line as a dashed line with underscore and space

    Loop us long as int i is less than the inputted answer variable

        Print i-i*10

endwhile


Start
Set start values
Loop
Print question
Get input
Print label
Print line
Så länge villkor inte är uppfyllt
Skriv ut rad

4. 3. Implementation and Test

```
if (rand_nr_a + rand_nr_b == answer) {
    printf("very good");
    stop = true;
} else {
    printf("No. Please try again.\n");
}
```

Here is the algorithm, I have tested with different testcases to make sure the sum is correct and compared with the answer.

For the second task I have an algorithm as well.

```
for (int i = 1; i<=answer; i++)
{
    printf("%d |\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",
           i, i, i*2, i*3, i*4, i*5, i*6, i*7, i*8, i*9, i*10);
}
```

As long as the counter is not equal to the inputed answer, it will print a new table.

I have tested the code with both high and low numbers.

5. 4. Results and discussion

In this section, you present the result and discuss on the result.

You can include what problems you have encountered and how they are solved. You should also include reflection on what you have learnt in this section. In this way, you improve your understanding and skills in programming.

6. 5. References

A reference list of works used in the report. Use the Vancouver system (https://en.wikipedia.org/wiki/Vancouver_system) in the listing.