Formative Task

MODULE CS1701 GROUP PROJECT LECTURES AND TUTORIALS

GROUP Red39 – TUTOR Dr Rumyana Neykova
MEMBERS |
ANA POLEAC -2004541
HAMNA AAMER -1945724
JENISHA PATEL -2010618
SIBE SINGH -1912122
ROHIL CHOKHANY -2043986

1.1 Requirements

The core functionalities must be included in all parts of the calculator project. The functional requirements are listed below; these requirements are what the program must do.

- User menu: displays the subtasks. Gives the user an option to select which subtask they want to use.

- User inputs: users should be able to input values and inputs they want. Each subtask should have a user input which allows the user to input values into the calculator. The program will not be user-friendly without this functionality.

- Exit or Continue: Another user input choice which can be included in the main menu so the user can exit the program once they are finished with the calculator.

- Each subtask should perform its calculations and display the results.

Sub-Task 2:

User Interface:

- A multiple-choice main menu, where users can choose to calculate between Power, Mod, and Root functions.

Erroneous user input:

- The user should be alerted for occasional error inputs in places such as the task menu.
- The program throws the user an error, either in a way of string or an error message with the program termination code.

Subtask Repeater:

- The user has the option to either restart the subtask or go back to the main menu at the end of the calculation. This repeater also has a validation for an erroneous input in case the user enters the wrong key.

Sub-Task 3:

User interface:

- User has a choice of whether they want to convert to metric units (Celsius) or imperial units (Fahrenheit)

Results displayed:

- For each calculation, the result should be displayed to the user

Continue or exit:

- Users can also exit the subtask if they are finished with their calculations

Validations:

- Input errors should not be accepted by the code

Sub-Task 4:

User interface:

- The user has a choice of what conversion they want and when to end the program as it will loop until they terminate it.
- Results of each calculation should be displayed after every calculation in a format the user is able to understand

Erroneous user input:

- To deal with this, extra if statements will have to be included as validations. E.g. The user input value for grams and centimeters can not be less than zero or string.

Continue or exit:

- The options , where the user chooses from weight or length, should keep looping until the user is done with every calculation they wanted to calculate.
- They should then have an option to return back to the main menu where they can exit.

Sub-Task 5:

User requirements:

- The user has to choose between using the option to convert from decimal to binary or other options implemented in the calculator.
- The user has to input a positive integer number, in which case the program will output the binary related to the input.

Erroneous user input :

- Users will receive a message that prompts them to input the right number.

Sub-Task 6:

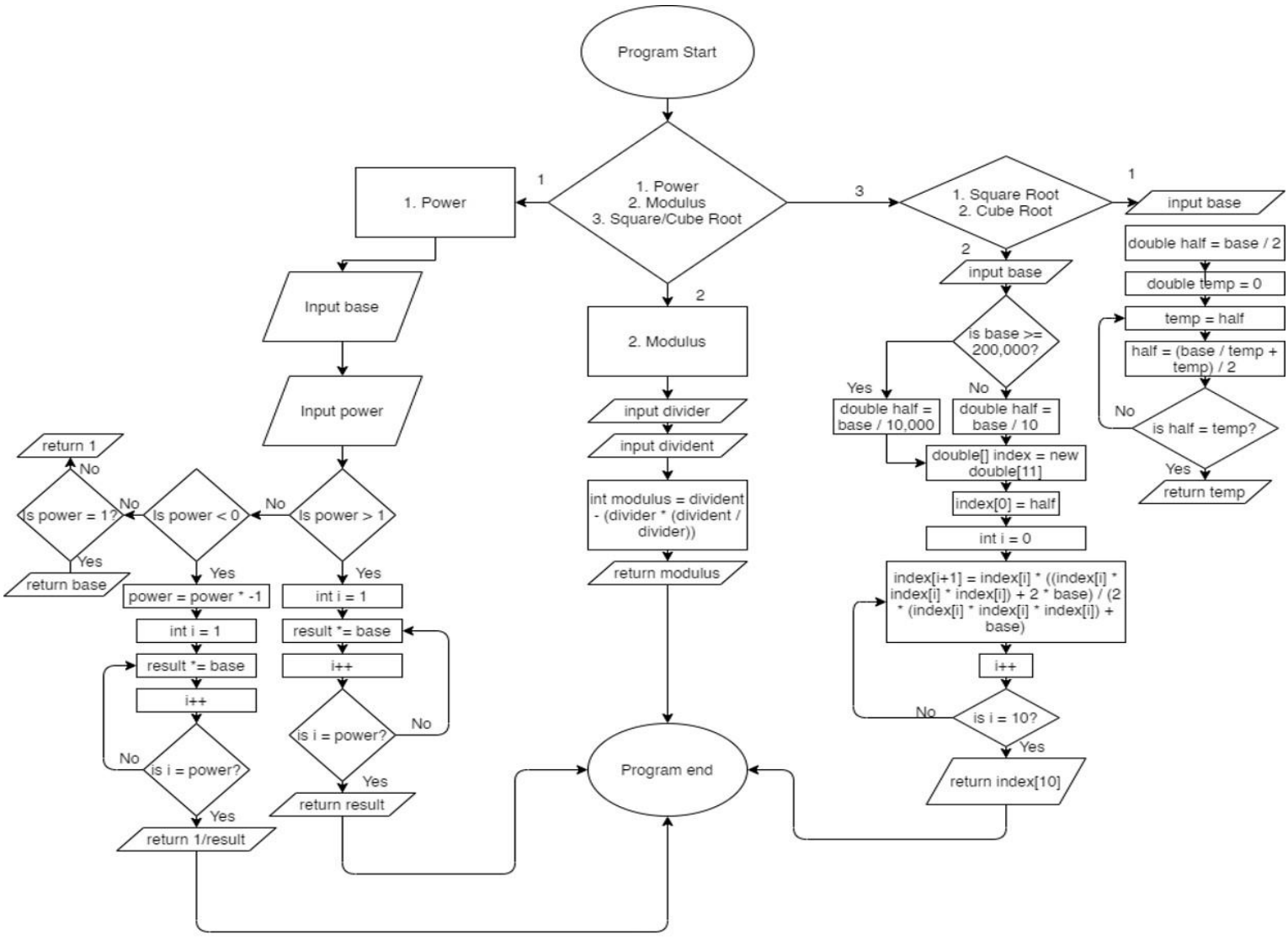
User requirements:

- The user is required to enter a binary number, which is later converted to a decimal.
- If the user selects task 6 from the main menu, they are asked to input a binary number which is then converted to a decimal number (e.g. 10110 = 22)

2.2 Task 2

Task 2: Creating a Power, Mod, and Square/Cube Root function without the use of given operation built-ins in Java. Provided below is a Flowchart and Pseudocode, with embedded features like user Input sections, error messages, and a Repeater that allows for the program to loop from the start.

2.2.1 Flowchart



2.2.2 Pseudocode

Main Menu

1) Output "1. Power function, 2. Mod function, 3. Square/Cube Root"
2) Read User Input
3) Input Result
4) If input = 1
    User Input Base
    User Input Power
    Result = PowerFunction (Base, Power)
    Output result
    Call Repeater
5) Else If Input = 2
    User Input Divider
    User Input Dividend
    Result = ModFunction (Divider, Dividend)
    Output Result
    Call Repeater
6) Else If Input = 3
    User Input Base
    Output "1. Square Root, 2. Cube Root"
    Read User Input2
    If Input 2 = 1
        Output SquareRoot(base)
    Else If input 2 = 2
        Output CubeRoot(base)
    Call Repeater
7) Else
    Throw Error Message
    Call Repeater

Power Function

1) User Input base
2) User Input power
3) Input result = base
4) If Power > 1
    I = 1
    Repeat while I < power
        Result = result * base
        I = I + 1
    Output result
5) If Power < 0
    Power = Power * -1
    I = 1
    Repeat while I < power
        Result = result * base
        I = I + 1
    Output 1/result
6) If power = 1
    Output base
7) Otherwise, output 1
8) Call Repeater

Mod Function

1) User Input Divider
2) User Input Dividend
3) Input modu = dividend - (divider * (dividend / divisor))
4) Output modu
5) Call Repeater

Square Root

1) User Input Base
2) Input Half = Base / 2
3) Input temp = 0
4) Repeat while half doesn't equal temp
    Temp = half
    Half = (base / temp + temp) / 2
5) Output temp
6) Call Repeater

Cube Root

1) User Input Base
2) Input Half
3) If base >= 200,000
    Half = base / 10,000
Else
    Half = base / 10
4) Input index array of length 11
5) Index [0] = half
6) For I = 0 to 9
    index[(i+1)] = index[i] * ((index[i] * index[i] * index[i]) + 2 * base) / (2 * (index[i] * index[i] * index[i]) + base)
    I = I + 1
7) Output Index [10]
8) Call Repeater

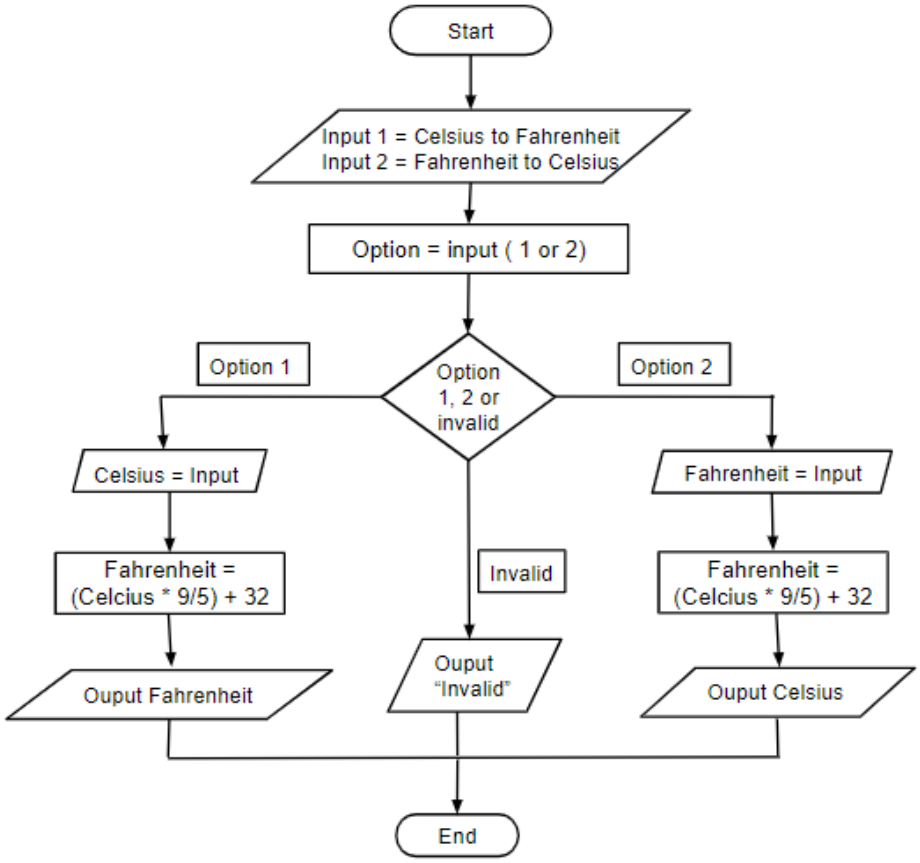
Repeater

1) Output "1. Go back to the Main Menu. 2. Exit the Program."
2) Read User Input
3) If Input = 1
    Call Main Menu
4) Else If Input = 2
    End Program
5) Else
    Throw Error Message
    6) End Program.

2.3 Task 3

Task 3: This task of the calculator converts temperature units between Celsius and Farenheit.

2.3.1 Flowchart



2.3.2 Pseudocode

The Pseudocode and the flowchart have been attached for the temperature converter from Celsius to Fahrenheit and vice versa.

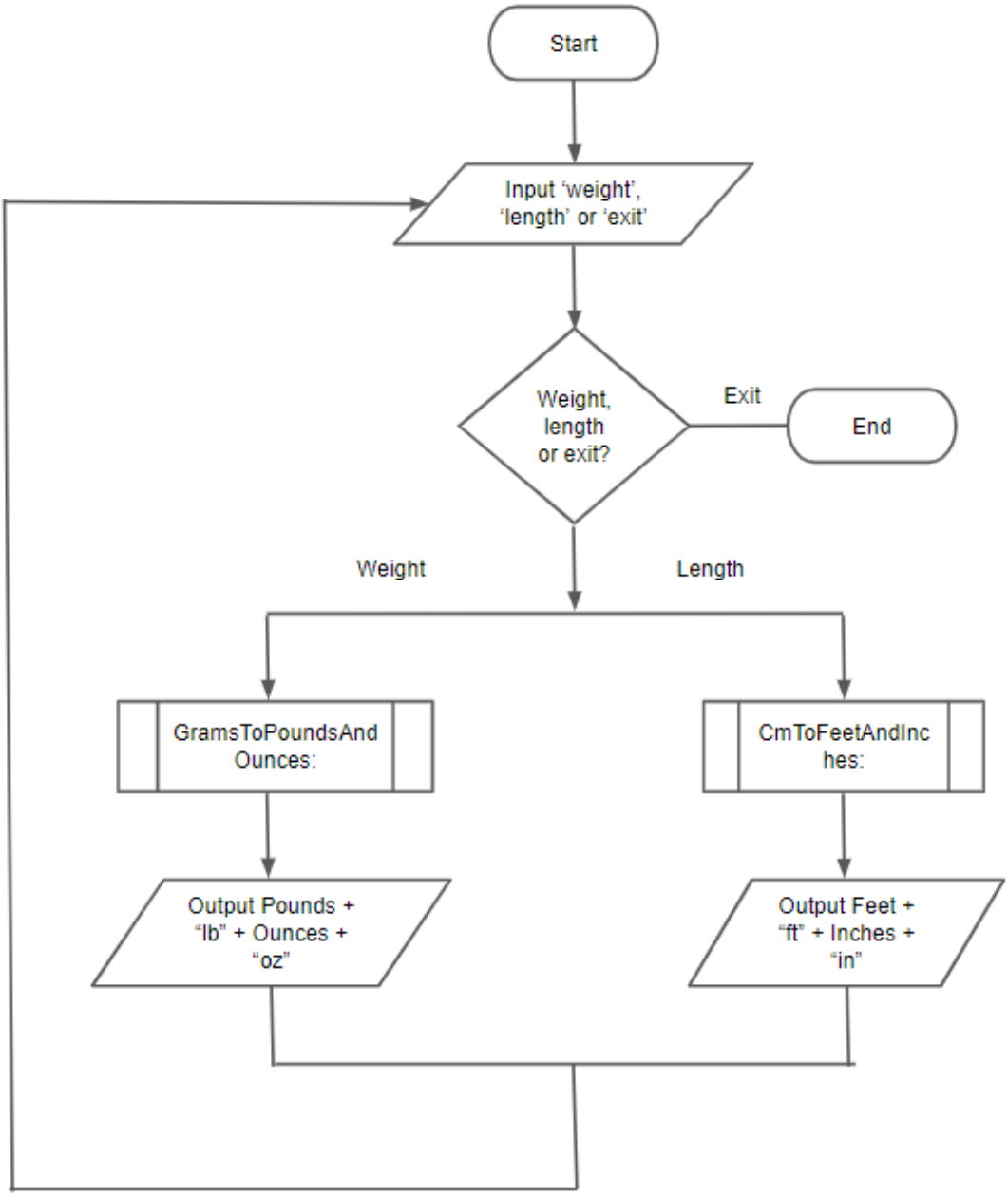
Algorithm 3: ConversionBetweenTemperatureUnits

1) Read User Input
2) If option = 1 (Celsius Input)
   Input = Celsius
   Fahrenheit = (Celsius * 9/5) + 32
   Output Print: "Temperature" + Fahrenheit + "Degrees Fahrenheit"
3) If option = 2 (Fahrenheit Input)
   Input = Fahrenheit
   Celsius = (Fahrenheit – 32) * 5/9
   Output Print: "Temperature" + Celsius + "Degrees Celsius"
4) If neither
   Display "Invalid"
5) End

2.4 Task 4

Task 4: This task of the calculator converts metric mass and length units to imperial units.

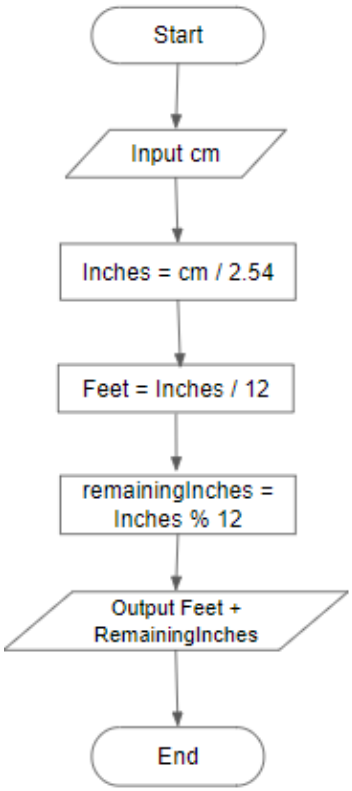
2.4.1 Flowchart



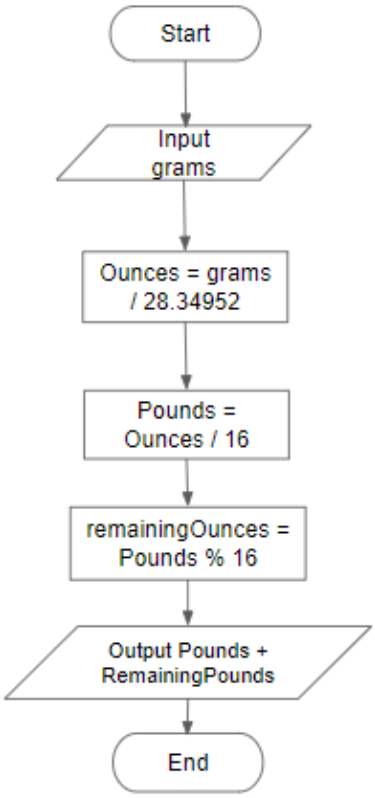
The above flow chart contains two sub processes called 'GramsToPoundsAndOunces' and 'CmToFeetAndInches'. The flow chart for the sub processes has been included below which describe the process of both sections.

2.4.2 Subprocesses

CmToFeetAndInches:



GramsToPoundsAndOunces:



2.4.3 Pseudocode

Algorithm 4- MetricToImperialForMassAndLength:

1) Read User input: 'Weight', 'Length' or 'Exit'.
2) If (Exit) #if the user picks the option Exit.
   Exit program
3) Else if (Weight) #if the user picks the option Weight.
   Input grams
   Ounces = grams/28.34952
   Pounds = Ounces / 16
   RemainingOunces = Pounds % 16
4) Output: Print (Pounds + "lb" + RemainingPounds + "oz")
5) Else if (Length) #if the user picks the option Length.
   Input cm
   Inches = cm / 2.54
   Feet = Inches / 12
   RemainingInches = Inches % 12
6) Output: Print (Feet + "ft" + RemainingInches + "in")
7) Loop to step 1 until user picks 'Exit' option

This pseudocode shows 7 steps, within these steps, details of what the if statements should do are included.

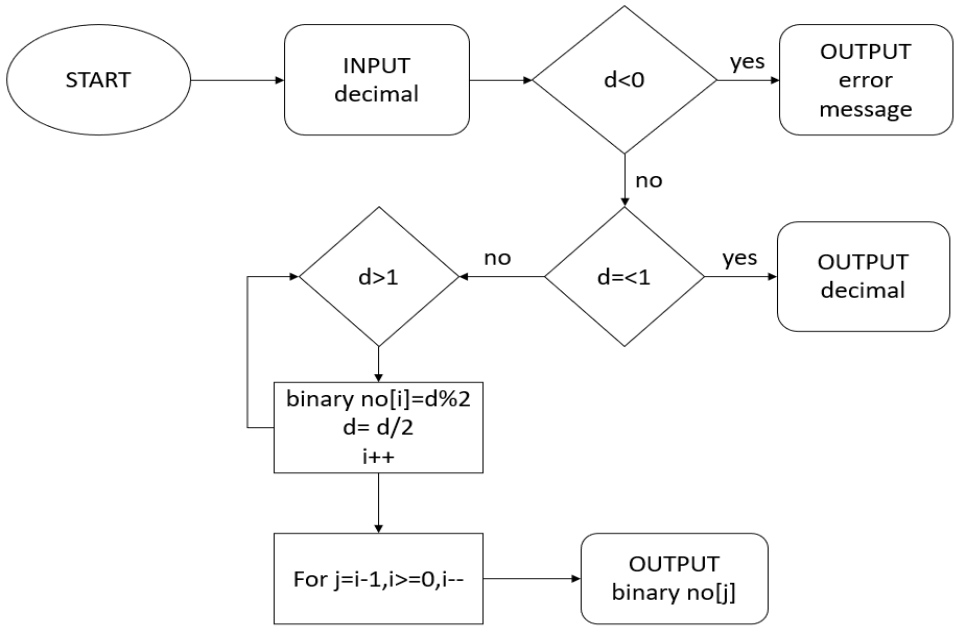
These designs only show how the code will work and does not show much of the user requirements and how it will be implemented into my code.

2.5 Task 5

Task 5: The conversion of a decimal number to a binary number.

2.5.1 Flowchart

The flowchart represents the clear steps that must be taken to convert a decimal number into a binary one.



2.5.2 Pseudocode

```
1)  Read user input
2)  IF (input < 0)
        Output error message
3)  ELSE IF (input <=1)
        Output decimal
4)  ELSE While (input>1)
        input array for binary number
        binary[i]=decimal%2
        decimal=decimal/2
        I++
5)  LOOP to step 4
6)  FOR index=i-1, index>=0, index--
7)  OUTPUT binary[index]
```

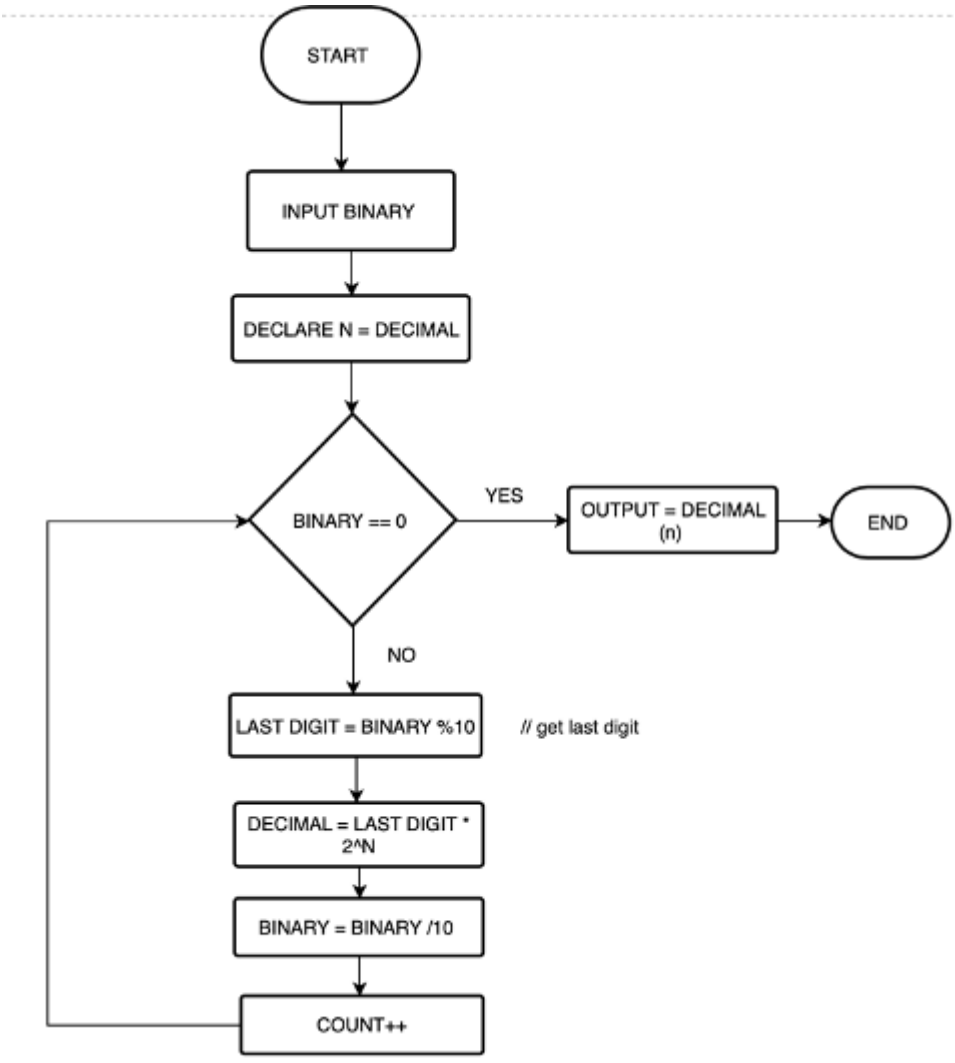
This pseudocode shows 7 steps which briefly present the method used to make the conversion from a decimal number to a binary one and how the code will work.

2.6 Task 6

Task 6 : This task converts a binary number to decimal number.

2.6.1 Flowchart

Flowchart are useful as they are a way of instant communication furthermore helps to increase efficiency.



2.6.2 Pseudocode

Algorithm 6: binarytoDecimal

Input: Binary (declare N, decimal)

```
1)  While binary =0
2)          *Display = Decimal*
3)Last digit = binary %10
4)              decimal = last digit *2^n
5)  Binary = binary/10
6) count++
7)      return to binary=0
```

Output: Decimal

Pseudocode is a great method for communicating ideas to other people to try and understand algorithms more effectively. This allows you to communicate complex ideas faster and clearer.

3.1 Code Integration

The final project can be found in the following repository:
https://github.com/rumineykova/assignment0-redgroup39