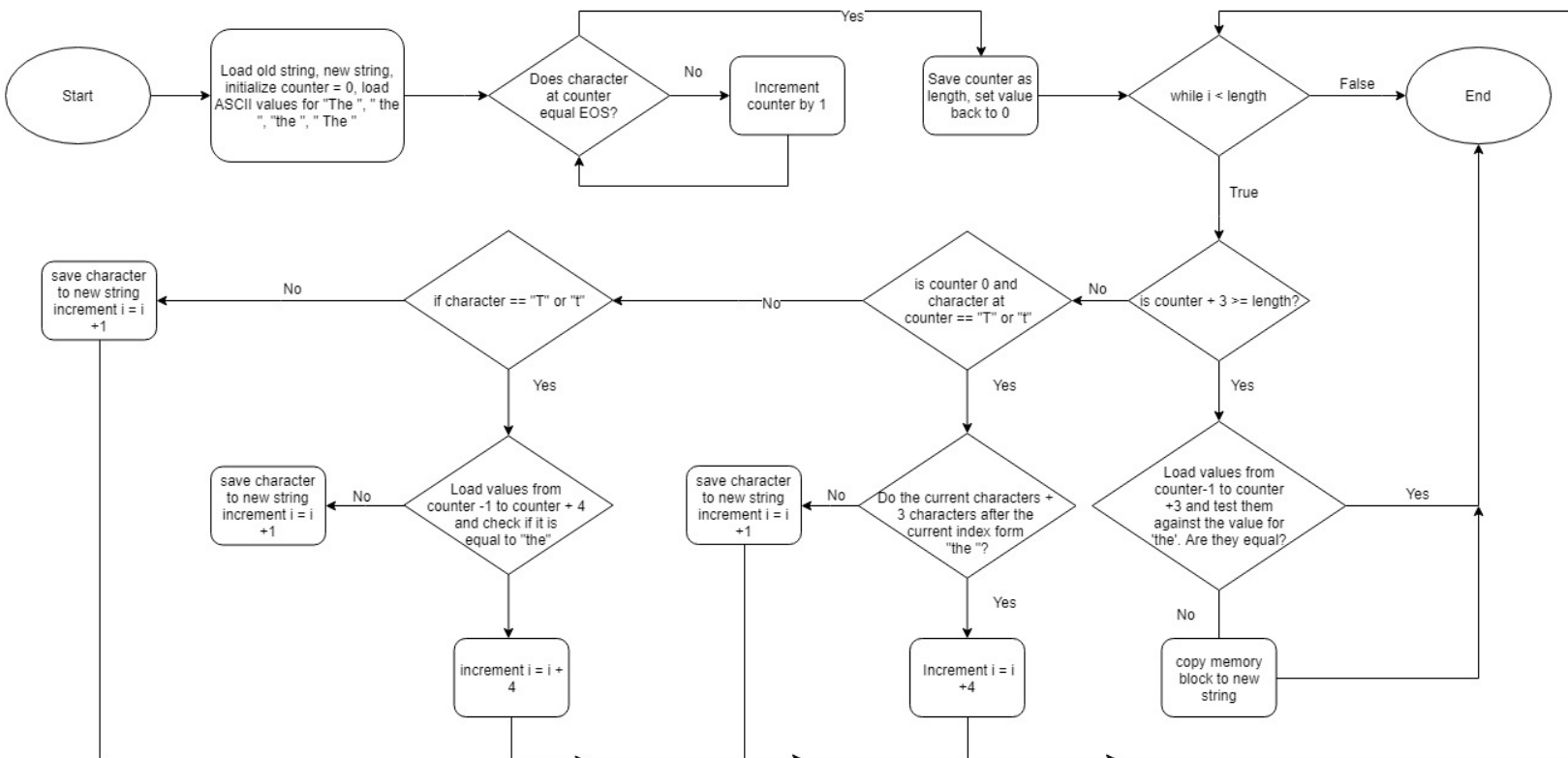


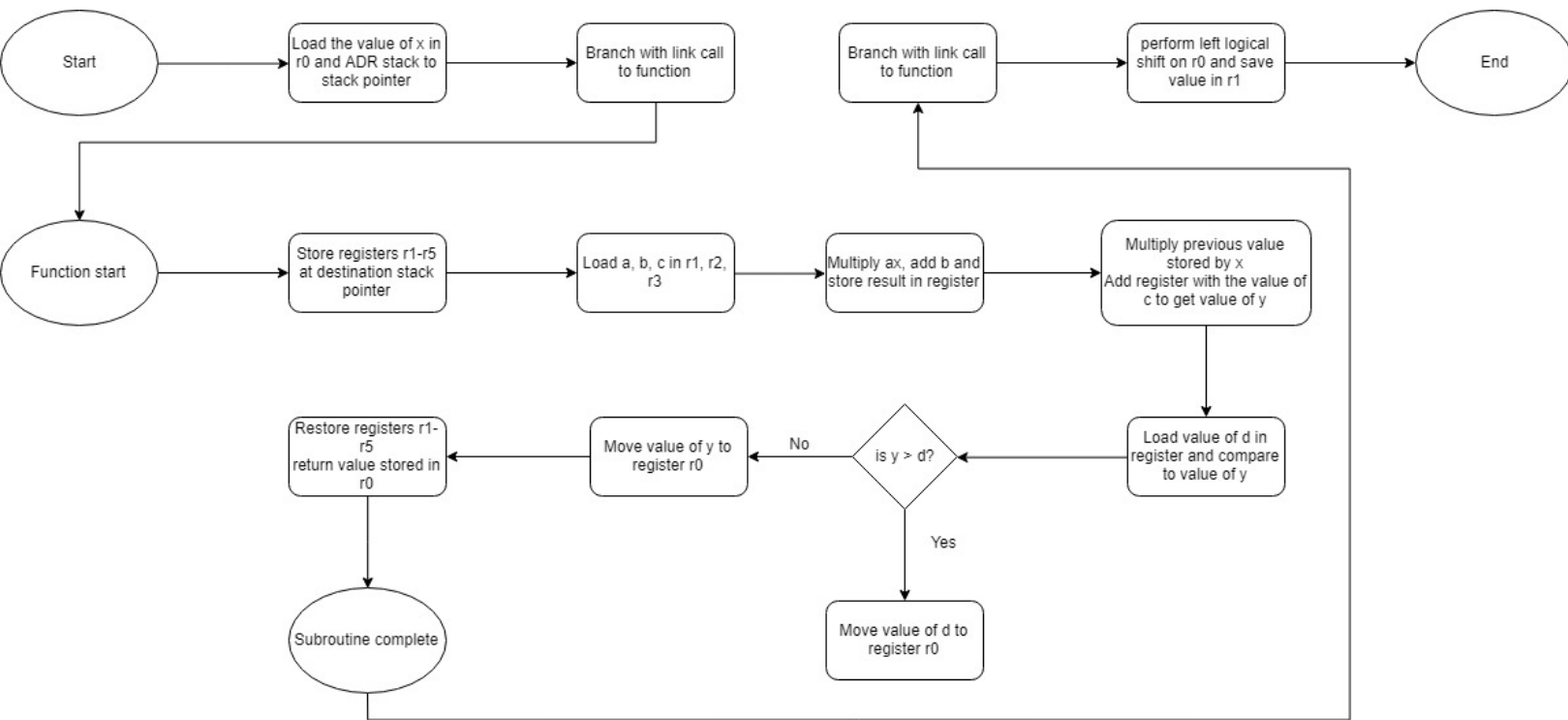
Question 1:



The program is coded in ARM Big Endian Processor using uVision4 IDE. The program is to validate String 1 into a new string called String 2. The program accepts a string of ASCII encoded characters. The program needs to copy the old string to a new string and eliminate all possible occurrences of the word 'the' from the old string.

The code is highly optimized to tackle the problem using as much memory and time as possible. The first string will be iterated, until the occurrence of the letter 't'. To validate if there exists the word 'the' originating from that letter 't', the program will check the character right before 't'. If that character is not a space or null character, meaning that the letter exists inside another word and hence cannot be a separate occurrence of the word 'the'. If there is a space or null character, the program then checks the next 3 characters against the word 'the'. If that is found true, the program counter increments by 4, without copying the word to the new string and resumes iteration.

Question 2:



The program is coded in ARM Big Endian Processor using uVision4 IDE. The program accepts 3 signed integer parameters and calculates the final value of an equation. 4 variables are defined using DCD assembly directives; a, b, c, d. The result is either the value d in r0, if $y > d$, otherwise the function will return y.

The subroutine is highly optimized using accessory registers, r1-r5, which are then restored at the of the subroutine.