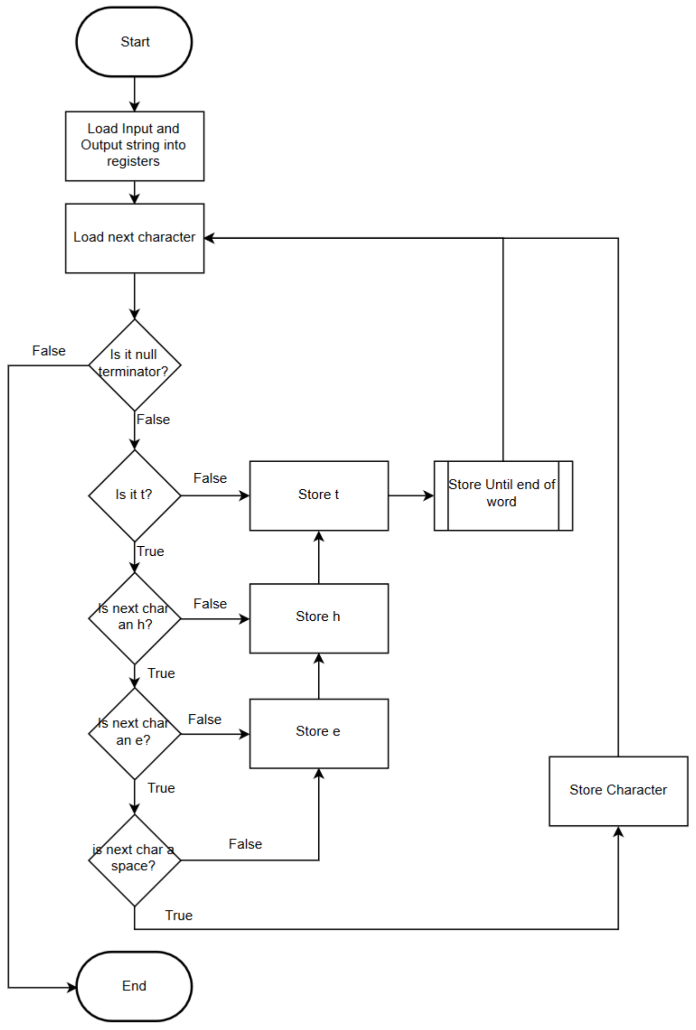
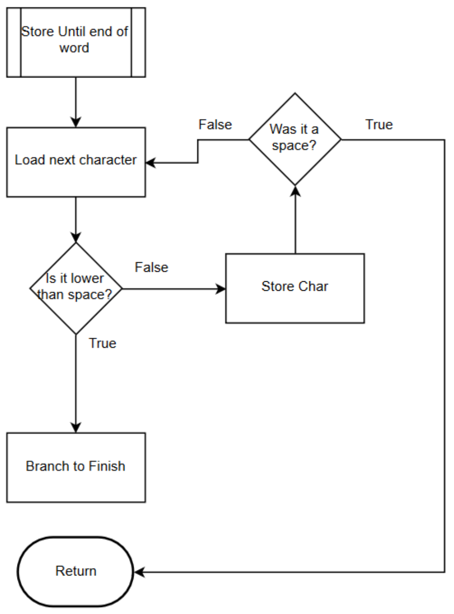
Report by: Johnathan Brunelle 250740613

Flowcharts

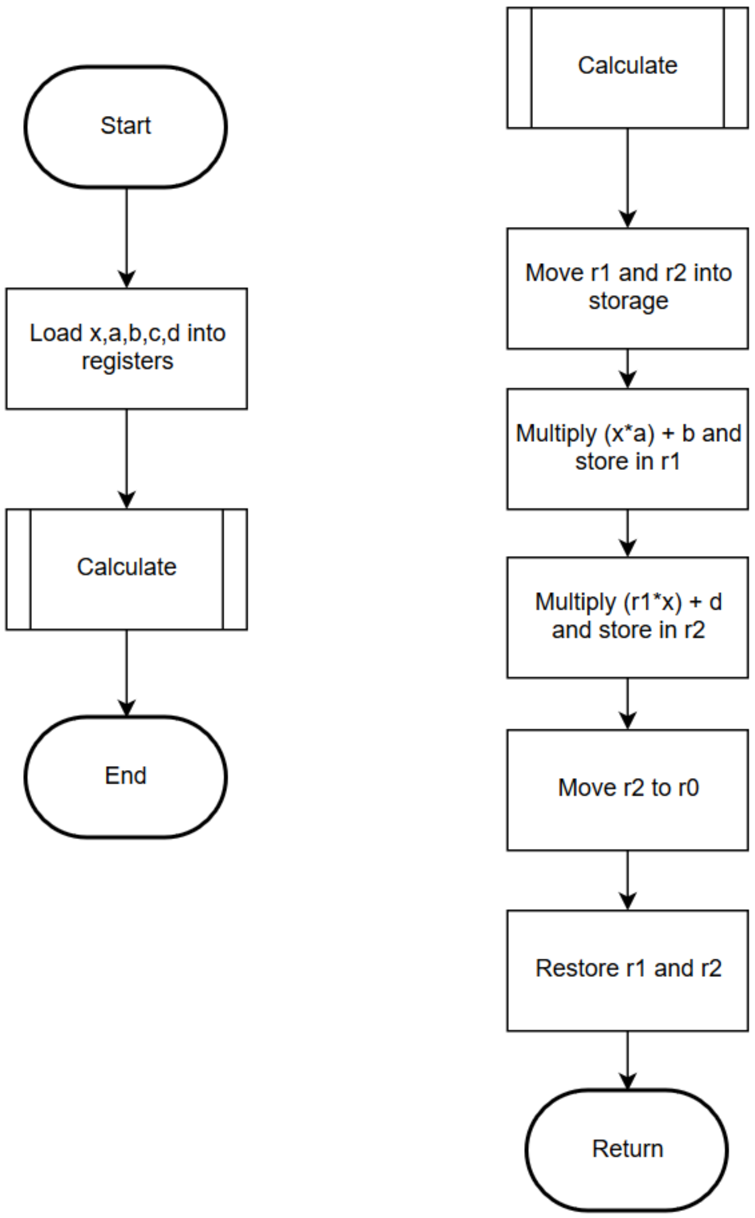
Question1:





Report by: Johnathan Brunelle 250740613

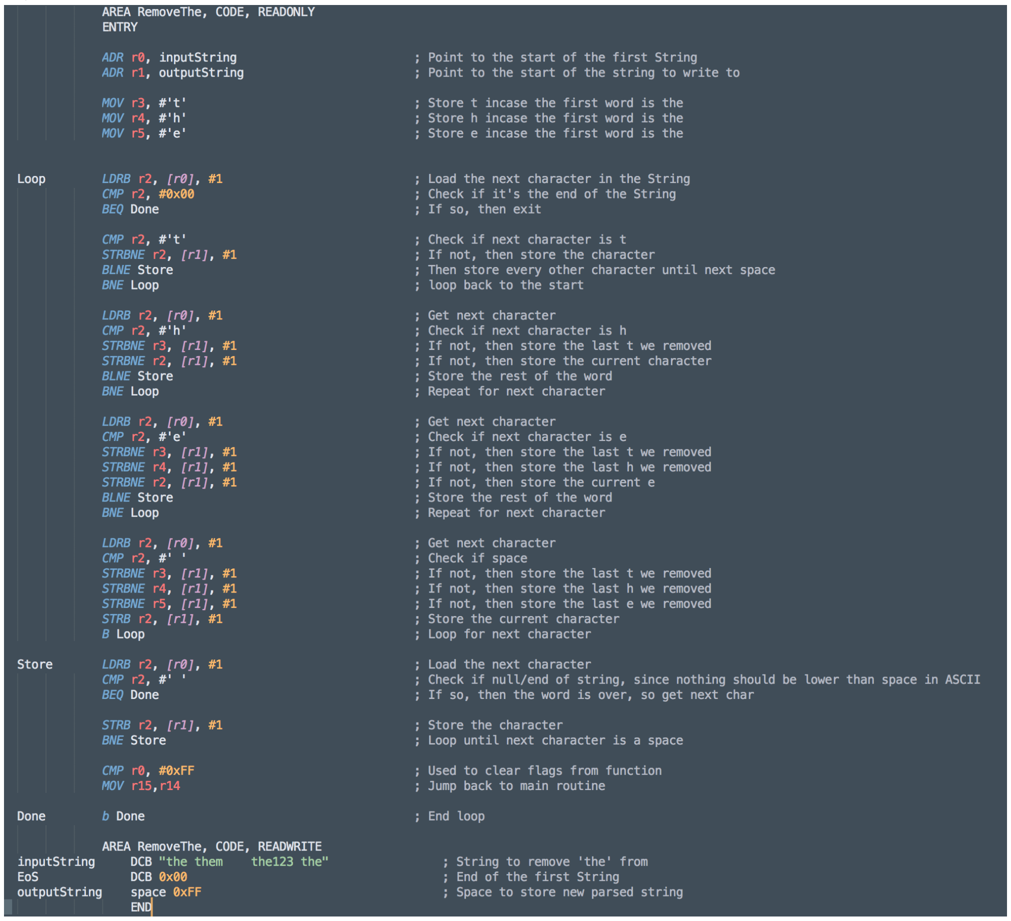
Question 2:


Start
Load x,a,b,c,d into registers
Calculate
End
Calculate
Move r1 and r2 into storage
Multiply (x*a) + b and store in r1
Multiply (r1*x) + d and store in r2
Move r2 to r0
Restore r1 and r2
Return

Report by: Johnathan Brunelle 250740613

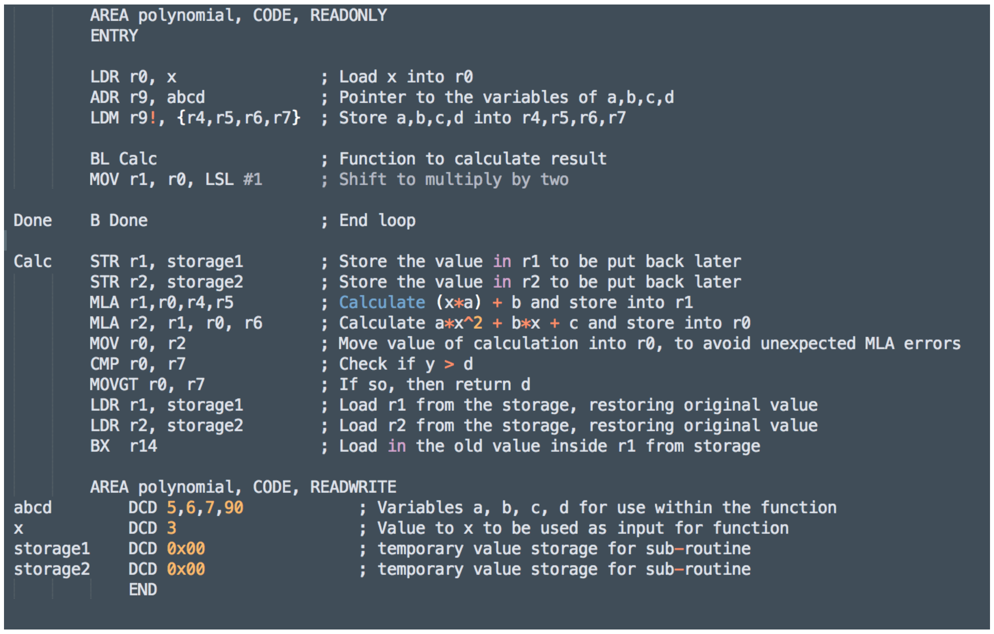
CODE:

Question 1:



```
            AREA RemoveThe, CODE, READONLY
            ENTRY

            ADR r0, inputString                         ; Point to the start of the first String
            ADR r1, outputString                        ; Point to the start of the string to write to

            MOV r3, #'t'                                ; Store t incase the first word is the
            MOV r4, #'h'                                ; Store h incase the first word is the
            MOV r5, #'e'                                ; Store e incase the first word is the


Loop        LDRB r2, [r0], #1                           ; Load the next character in the String
            CMP r2, #0x00                               ; Check if it's the end of the String
            BEQ Done                                    ; If so, then exit

            CMP r2, #'t'                                ; Check if next character is t
            STRBNE r2, [r1], #1                         ; If not, then store the character
            BLNE Store                                  ; Then store every other character until next space
            BNE Loop                                    ; loop back to the start

            LDRB r2, [r0], #1                           ; Get next character
            CMP r2, #'h'                                ; Check if next character is h
            STRBNE r3, [r1], #1                         ; If not, then store the last t we removed
            STRBNE r2, [r1], #1                         ; If not, then store the current character
            BLNE Store                                  ; Store the rest of the word
            BNE Loop                                    ; Repeat for next character

            LDRB r2, [r0], #1                           ; Get next character
            CMP r2, #'e'                                ; Check if next character is e
            STRBNE r3, [r1], #1                         ; If not, then store the last t we removed
            STRBNE r4, [r1], #1                         ; If not, then store the last h we removed
            STRBNE r2, [r1], #1                         ; If not, then store the current e
            BLNE Store                                  ; Store the rest of the word
            BNE Loop                                    ; Repeat for next character

            LDRB r2, [r0], #1                           ; Get next character
            CMP r2, #' '                                ; Check if space
            STRBNE r3, [r1], #1                         ; If not, then store the last t we removed
            STRBNE r4, [r1], #1                         ; If not, then store the last h we removed
            STRBNE r5, [r1], #1                         ; If not, then store the last e we removed
            STRB r2, [r1], #1                           ; Store the current character
            B Loop                                      ; Loop for next character

Store       LDRB r2, [r0], #1                           ; Load the next character
            CMP r2, #' '                                ; Check if null/end of string, since nothing should be lower than space in ASCII
            BEQ Done                                    ; If so, then the word is over, so get next char

            STRB r2, [r1], #1                           ; Store the character
            BNE Store                                   ; Loop until next character is a space

            CMP r0, #0xFF                               ; Used to clear flags from function
            MOV r15,r14                                 ; Jump back to main routine

Done        b Done                                      ; End loop

            AREA RemoveThe, CODE, READWRITE
inputString     DCB "the them    the123 the"                ; String to remove 'the' from
EoS             DCB 0x00                                    ; End of the first String
outputString    space 0xFF                                  ; Space to store new parsed string
                END
```

Question2:



```
        AREA polynomial, CODE, READONLY
        ENTRY

        LDR r0, x               ; Load x into r0
        ADR r9, abcd            ; Pointer to the variables of a,b,c,d
        LDM r9!, {r4,r5,r6,r7}  ; Store a,b,c,d into r4,r5,r6,r7

        BL Calc                 ; Function to calculate result
        MOV r1, r0, LSL #1      ; Shift to multiply by two

Done    B Done                  ; End loop

Calc    STR r1, storage1        ; Store the value in r1 to be put back later
        STR r2, storage2        ; Store the value in r2 to be put back later
        MLA r1,r0,r4,r5         ; Calculate (x*a) + b and store into r1
        MLA r2, r1, r0, r6      ; Calculate a*x^2 + b*x + c and store into r0
        MOV r0, r2              ; Move value of calculation into r0, to avoid unexpected MLA errors
        CMP r0, r7              ; Check if y > d
        MOVGT r0, r7            ; If so, then return d
        LDR r1, storage1        ; Load r1 from the storage, restoring original value
        LDR r2, storage2        ; Load r2 from the storage, restoring original value
        BX  r14                 ; Load in the old value inside r1 from storage

        AREA polynomial, CODE, READWRITE
abcd        DCD 5,6,7,90            ; Variables a, b, c, d for use within the function
x           DCD 3                   ; Value to x to be used as input for function
storage1    DCD 0x00                ; temporary value storage for sub-routine
storage2    DCD 0x00                ; temporary value storage for sub-routine
            END
```