

Electrical and Computer Engineering

ENCS2380 – Computer Organization and Microprocessors- Fall 2023 Course Project (2)

Student Name: Student ID NO:

Saif Al-Deen Battah 1170986

Abdulkailq injas 1173251

Instructor: Dr. Abualseoud Hanani.

Section NO: 1.

Date: 20/2/2023.

Contents

Part 1	3
Part 2	
Part 3	
Part 4	
Appendix	

```
1
 2
        PRESERVE8
 3
        THUMB
        AREA RESET, DATA, READONLY
 4
        EXPORT __Vectors
 5
 6
 7
        ;part 1
 8
 9
            AREA
                   String, DATA, READWRITE
            sentence1 DCB "Arm Assembly Programming", 0
10
11
            sentence2 DCB "Keil uVision 5 Software", 0
12
```

```
13 ;part 2
14
15
         AREA
                 Convert, CODE, READONLY
16
17
            ; Declare the procedure
18
            CNV_CASE
19
            EXPORT CNV_CASE
20
            CNV CASE
21
            PUSH
                   {LR}
22
           MOV
                   r2, #0
                                  ; Count of converted letters
23
        loop
24
            LDRB
                   r3, [r0], #1
                                  ; Load a byte from the string
25
            CMP
                   r3, #0
                                  ; Check for end of string
26
            BEQ
                   end
                   r3, #'A'
                                  ; Check if the character is a capital letter
27
            CMP
28
            BLT
                   not capital
29
            CMP
                   r3, #'Z'
30
            BGT
                   not capital
31
            ADD
                   r3, r3, #32
                                  ; Convert to lowercase
32
            ADD
                   r2, r2, #1
                                  ; Increment count of converted letters
33
        not capital
            STRB
34
                   r3, [rl], #1
                                  ; Store the converted byte in the new string
35
            В
                   loop
36
        end
37
           MOV
                   r0, r2
                                  ; Return the count of converted letters in RO
38
           POP
                   {PC}
39
            ; Convert the first string to lowercase
40
41
            AREA
                   TXT1, DATA, READWRITE
            CNV CASE
42
                   rO, =sentencel ; Load the address of the first string
43
            LDR
                             ; Load the address of the new string
                   rl, =TXTl
44
            LDR
45
            BL
                   CNV_CASE
                                  ; Call the procedure
                     CNV CASE
             BL
                                  ; Call the procedure
 45
                                  ; Load the address of the count variable
 46
             LDR
                     rl, =Countl
 47
                                     ; Store the count of converted letters
             STR
                     r0, [r1]
 48
             ; Convert the second string to lowercase
 49
             AREA
 50
                     TXT2, DATA, READWRITE
             CNV CASE
 51
 52
             LDR
                     rO, =sentence2 ; Load the address of the second string
53
             LDR
                     rl_r = TXT2
                                     ; Load the address of the new string
54
             BL
                     CNV CASE
                                     ; Call the procedure
 55
             LDR
                     rl, =Count2
                                    ; Load the address of the count variable
 56
             STR
                     r0, [r1]
                                     ; Store the count of converted letters
```

```
58
             ;part 3
                      Common, DATA, READWRITE
  59
             AREA
  60
  61
             ; Declare the procedure
             COMMON_CHARS
  62
             EXPORT COMMON_CHARS
  63
             COMMON_CHARS
  64
  65
             PUSH
                   {LR}
                    r2, #0
  66
             MOV
                                    ; Counter for common characters
  67
             MOV
                    r4, #0
                                    ; Index for string 2
  68
          loopl
  69
                    r3, [r0], #1
                                   ; Load a byte from the first string
  70
             CMP
                    r3, #0
                                    ; Check for end of string
             BEO
  71
                    end
  72
             MOV
                    rl, #0
                                    ; Index for string 2
  73
          loop2
  74
             LDRB
                    r5, [r1, r5]
                                   ; Load a byte from the second string
  75
             CMP
                    r5, #0
                                    ; Check for end of string
  76
                    end2
             BEQ
  77
                    r5, #32
                                   ; Convert to lowercase if it's a capital letter
             CMP
  78
             BLT
                    not_capital2
  79
             CMP
                    r5, #'Z'
  80
             BGT
                    not_capital2
  81
             ADD
                    r5, r5, #32
  82
          not capital2
  83
             CMP
                    r3, r5
                                    ; Compare the characters
  84
             BNE
                     not_common
  85
             ADD
                    r2, r2, #1
                                   ; Increment the counter
  86
             В
                     end2
  87
          not_common
             ADD
  88
                    rl, rl, #1
                                   ; Increment the index for string 2
  89
             В
                     loop2
 89
                       loop2
 90
          end2
 91
               ADD
                       r4, r4, #1
                                          ; Increment the index for string 1
 92
               MOV
                       r5, #0
                                          ; Reset the index for string 2
 93
               В
                        loopl
 94
          end
 95
               MOV
                       r0, r2
                                          ; Return the counter in RO
 96
               POP
                        {PC}
 97
 98
               ; Compute the number of common characters between the two strings
99
                        COMMON, DATA, READWRITE
               AREA
100
               COMMON CHARS
101
               LDR
                       r0, =TXT1
                                         ; Load the address of the first string
102
                                         ; Load the address of the second string
               LDR
                        rl, =TXT2
                        COMMON CHARS
                                         ; Call the procedure
103
               BL
104
                       rl, =COMMON
                                         ; Load the address of the counter variable
               LDR
105
               STR
                       r0, [r1]
                                          ; Store the counter
106
107
108
```

```
107 ; part 4
 108
                                                  ; Shift the carry bit into bit 0
 109
                                 r3, r3, #1
                          r3, r3, #1 ; Invert bit 0
r3, [r2], #1 ; Store the encrypted byte and increment the destination pointer
 110
                 BIC
 111
                 STRB
 112
                 CMP
                          r3, #0
                                            ; Check for end of string
                BNE
 113
                          loop
 114
                POP
                          {LR}
 115
 116
                 ; Encrypt the first string
 117
                 AREA Encryption, DATA, READWRITE
 118
                 ENCRYPT_STRING1
                          rO, =TXTl ; Load the address of the first string rl, =ENCRYPTl ; Load the address of the destination array
 119
                 LDR
                          r0, =TXT1
 120
                 LDR
                          ENCRYPT_STRING ; Call the encryption procedure
 121
                 BL
 122
 123
                 ; Encrypt the second string
                 ENCRYPT_STRING2
 124
                          r0, =TXT2 ; Load the address of the second string
r1, =ENCRYPT2 ; Load the address of the destination array
ENCRYPT_STRING ; Call the encryption procedure
 125
                 LDR
                         r0, =TXT2
 126
                 LDR
 127
                 BL
 128
 129
130
```

Appendix

```
PRESERVE8
   THUMB
   AREA RESET, DATA, READONLY
    EXPORT __Vectors
   ;part 1
         AREA String, DATA, READWRITE
 sentence1 DCB "Arm Assembly Programming",0
 sentence2 DCB "Keil uVision 5 Software",0
;part 2
   AREA Convert, CODE, READONLY
 ; Declare the procedure
 CNV_CASE
 EXPORT CNV_CASE
 CNV CASE
 PUSH {LR}
 MOV r2, #0 ; Count of converted letters
loop
 LDRB r3, [r0], #1; Load a byte from the string
```

```
r3, #0; Check for end of string
  CMP
  BEQ
        end
  CMP
        r3, #'A'
                  ; Check if the character is a capital letter
       not_capital
  BLT
       r3, #'Z'
  CMP
  BGT
        not capital
        r3, r3, #32 ; Convert to lowercase
  ADD
       r2, r2, #1; Increment count of converted letters
  ADD
not capital
 STRB r3, [r1], #1; Store the converted byte in the new string
  В
      loop
end
  MOV r0, r2; Return the count of converted letters in R0
  POP
        {PC}
  ; Convert the first string to lowercase
  AREA TXT1, DATA, READWRITE
 CNV CASE
       r0, =sentence1; Load the address of the first string
 LDR
       r1, =TXT1 ; Load the address of the new string
       CNV CASE ; Call the procedure
  BL
      r1, =Count1; Load the address of the count variable
  LDR
  STR
        r0, [r1]; Store the count of converted letters
```

```
; Convert the second string to lowercase
  AREA TXT2, DATA, READWRITE
 CNV_CASE
       r0, =sentence2; Load the address of the second string
  LDR
  LDR
       r1, =TXT2; Load the address of the new string
       CNV_CASE ; Call the procedure
  BL
       r1, =Count2 ; Load the address of the count variable
  LDR
  STR
                 ; Store the count of converted letters
       r0, [r1]
         ;part 3
         AREA
                Common, DATA, READWRITE
 ; Declare the procedure
  COMMON CHARS
  EXPORT COMMON_CHARS
  COMMON_CHARS
  PUSH {LR}
                  ; Counter for common characters
  MOV
        r2, #0
                  ; Index for string 2
  MOV
        r4, #0
loop1
 LDRB r3, [r0], #1; Load a byte from the first string
                  ; Check for end of string
  CMP
        r3, #0
  BEQ
        end
                  ; Index for string 2
  MOV r1, #0
```

```
loop2
        r5, [r1, r5]; Load a byte from the second string
  LDRB
                   ; Check for end of string
  CMP
        r5, #0
        end2
  BEQ
        r5, #32; Convert to lowercase if it's a capital letter
  CMP
       not_capital2
  BLT
        r5, #'Z'
  CMP
        not_capital2
  BGT
        r5, r5, #32
  ADD
not capital2
                  ; Compare the characters
  CMP
        r3, r5
  BNE
        not_common
        r2, r2, #1 ; Increment the counter
  ADD
  В
      end2
not_common
        r1, r1, #1 ; Increment the index for string 2
  ADD
  В
      loop2
end2
        r4, r4, #1 ; Increment the index for string 1
  ADD
 MOV r5, #0
                   ; Reset the index for string 2
      loop1
  В
end
               ; Return the counter in RO
  MOV r0, r2
      {PC}
  POP
```

```
; Compute the number of common characters between the two strings
    AREA COMMON, DATA, READWRITE
    COMMON CHARS
    LDR
          r0, =TXT1 ; Load the address of the first string
    LDR
          r1, =TXT2; Load the address of the second string
         COMMON_CHARS ; Call the procedure
    BL
         r1, =COMMON ; Load the address of the counter variable
    LDR
    STR
          r0, [r1]
                    ; Store the counter
   ; part 4
           LSL r3, r3, #1; Shift the carry bit into bit 0
    BIC
         r3, r3, #1; Invert bit 0
          r3, [r2], #1; Store the encrypted byte and increment the destination
pointer
    CMP
                 ; Check for end of string
         r3, #0
    BNE
          loop
    POP
          {LR}
    ; Encrypt the first string
    AREA Encryption, DATA, READWRITE
    ENCRYPT STRING1
          r0, =TXT1 ; Load the address of the first string
    LDR
         r1, =ENCRYPT1; Load the address of the destination array
    LDR
```

BL ENCRYPT_STRING; Call the encryption procedure

; Encrypt the second string

ENCRYPT_STRING2

LDR r0, =TXT2 ; Load the address of the second string

LDR r1, =ENCRYPT2 ; Load the address of the destination array

BL ENCRYPT_STRING ; Call the encryption procedure