ARM ASSEMBLY LAB-4

By

Swapneel Pimparkar (CS18M516)
CS6620a
Prof. Madhu Mutyam

STATUS UPDATE

The ARM assembly code was written for 32 bit processor and verified using ARMSim simulator successfully.

EXERCISE – COMPARE TWO STRINGS OF ASCII CHARACTERS TO SEE WHICH IS LARGER.

Compare two strings of ASCII characters to see which is larger (i.e., which follows the other in alphabetical ordering). Both strings have the same length as defined by the LENGTH variable. The strings' starting addresses are defined by the START1 and START2 variables. If the string defined by START1 is greater than or equal to the other string, clear the GREATER variable; otherwise set the variable to all ones (0xFFFFFFFF).

The logic to solve this exercise is hand-coded in 32 bit ARM Assembly and verified on ARMSim Simulator.

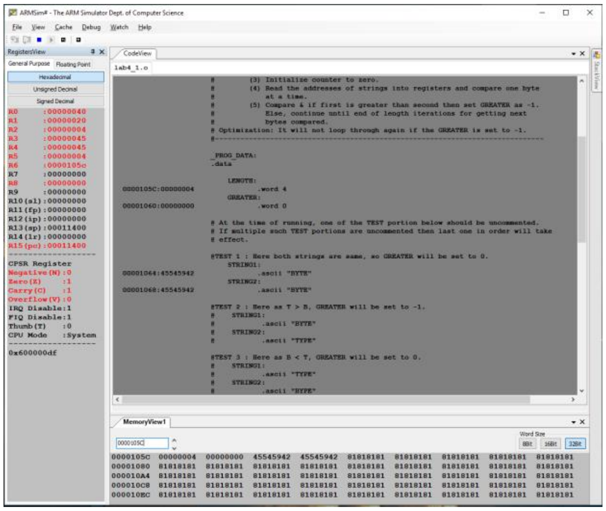
The logic used is mentioned in the code file itself and all the necessary instructions are supplied with comments.

Various unit tests are also part of the code file itself. Need to uncomment and run.

All the unit tests passed (various values of STRINGs).

SCREENSHOT - TEST 1

The sample output screenshot for the logic is – (Registers listed on the left are to be noted. All are zero to begin with). Similarly, the memory view in bottom pane of the Simulator Window. Output address (where GREATER is stored) is – 0x00001060.

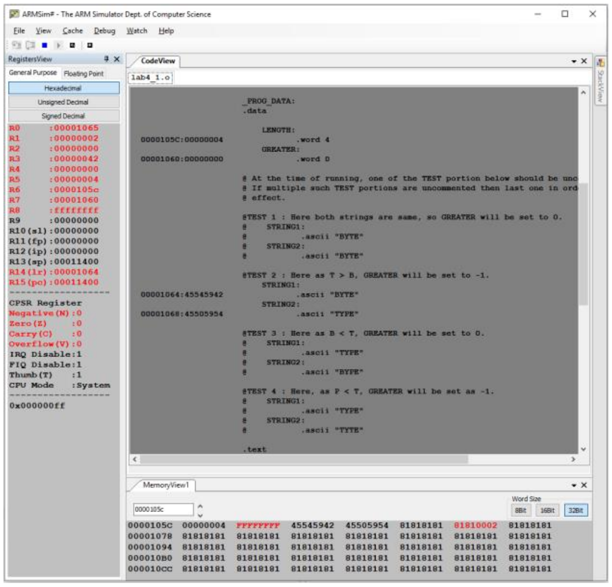

ARMSim# - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
CodeView
lab4_1.o
MemoryView1
Word Size
8Bit 16Bit 32Bit
StackView

```
R0       :00000040
R1       :00000020
R2       :00000004
R3       :00000045
R4       :00000045
R5       :00000004
R6       :0000105c
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00000000
R15(pc)  :00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

```
                        @         (3) Initialize counter to zero.
                        @         (4) Read the addresses of strings into registers and compare one byte
                        @             at a time.
                        @         (5) Compare & if first is greater than second then set GREATER as -1.
                        @             Else, continue until end of length iterations for getting next
                        @             bytes compared.
                        @ Optimization: It will not loop through again if the GREATER is set to -1.
                        @-----------------------------------------------------------------------------

                        _PROG_DATA:
                        .data

                            LENGTH:
0000105C:00000004                   .word 4
                            GREATER:
00001060:00000000                   .word 0

                        @ At the time of running, one of the TEST portion below should be uncommented.
                        @ If multiple such TEST portions are uncommented then last one in order will take
                        @ effect.

                        @TEST 1 : Here both strings are same, so GREATER will be set to 0.
                            STRING1:
00001064:45545942                   .ascii "BYTE"
                            STRING2:
00001068:45545942                   .ascii "BYTE"

                        @TEST 2 : Here as T > B, GREATER will be set to -1.
                        @    STRING1:
                        @            .ascii "BYTE"
                        @    STRING2:
                        @            .ascii "TYPE"

                        @TEST 3 : Here as B < T, GREATER will be set to 0.
                        @    STRING1:
                        @            .ascii "TYPE"
                        @    STRING2:
                        @            .ascii "BYPE"
```

```
0000105C
0000105c  00000004  00000000  45545942  45545942  81818181  81818181  81818181  81818181  81818181
00001080  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010A4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010C8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010EC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

SCREENSHOT - TEST 2

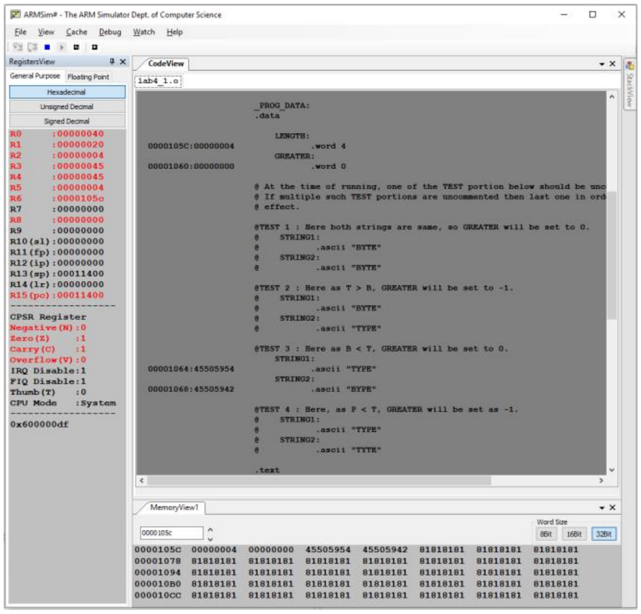

ARMSim# - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
CodeView
lab4_1.o
MemoryView1
Word Size
8Bit 16Bit 32Bit
StackView

```
R0       :00001065
R1       :00000002
R2       :00000000
R3       :00000042
R4       :00000000
R5       :00000004
R6       :0000105c
R7       :00001060
R8       :ffffffff
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00011400
R14(lr)  :00001064
R15(pc)  :00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :1
CPU Mode   :System
------------------
0x000000ff
```

```
                        _PROG_DATA:
                        .data

                            LENGTH:
0000105C:00000004                   .word 4
                            GREATER:
00001060:00000000                   .word 0

                        @ At the time of running, one of the TEST portion below should be unc
                        @ If multiple such TEST portions are uncommented then last one in ord
                        @ effect.

                        @TEST 1 : Here both strings are same, so GREATER will be set to 0.
                        @    STRING1:
                        @            .ascii "BYTE"
                        @    STRING2:
                        @            .ascii "BYTE"

                        @TEST 2 : Here as T > B, GREATER will be set to -1.
                            STRING1:
00001064:45545942                   .ascii "BYTE"
                            STRING2:
00001068:45505954                   .ascii "TYPE"

                        @TEST 3 : Here as B < T, GREATER will be set to 0.
                        @    STRING1:
                        @            .ascii "TYPE"
                        @    STRING2:
                        @            .ascii "BYPE"

                        @TEST 4 : Here, as P < T, GREATER will be set as -1.
                        @    STRING1:
                        @            .ascii "TYPE"
                        @    STRING2:
                        @            .ascii "TYTE"

                        .text
```

```
0000105c
0000105C  00000004  FFFFFFFF  45545942  45505954  81818181  81810002  81818181
00001078  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001094  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010CC  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

SCREENSHOT - TEST 3

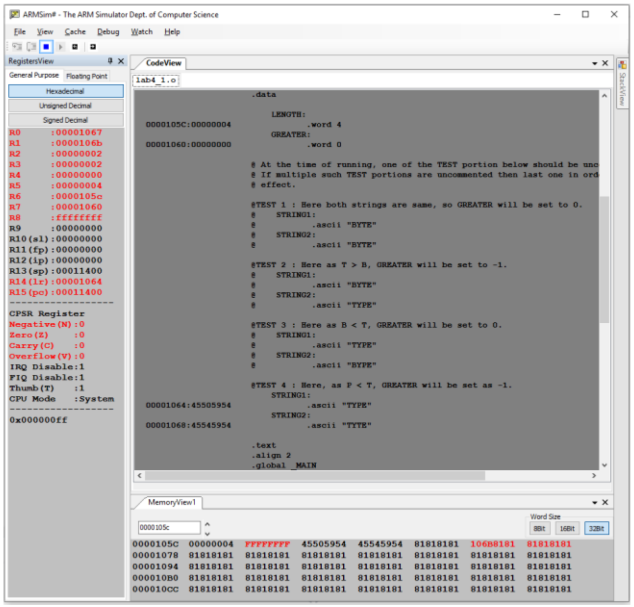


```
R0        :00000040
R1        :00000020
R2        :00000004
R3        :00000045
R4        :00000045
R5        :00000004
R6        :0000105c
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):00011400
-------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
-------------------
0x600000df
```

```
                        _PROG_DATA:
                        .data

                            LENGTH:
0000105C:00000004                   .word 4
                            GREATER:
00001060:00000000                   .word 0

                        @ At the time of running, one of the TEST portion below should be unc
                        @ If multiple such TEST portions are uncommented then last one in ord
                        @ effect.

                        @TEST 1 : Here both strings are same, so GREATER will be set to 0.
                        @    STRING1:
                        @            .ascii "BYTE"
                        @    STRING2:
                        @            .ascii "BYTE"

                        @TEST 2 : Here as T > B, GREATER will be set to -1.
                        @    STRING1:
                        @            .ascii "BYTE"
                        @    STRING2:
                        @            .ascii "TYPE"

                        @TEST 3 : Here as B < T, GREATER will be set to 0.
                            STRING1:
00001064:45505954                   .ascii "TYPE"
                            STRING2:
00001068:45505942                   .ascii "BYPE"

                        @TEST 4 : Here, as P < T, GREATER will be set as -1.
                        @    STRING1:
                        @            .ascii "TYPE"
                        @    STRING2:
                        @            .ascii "TYTE"

                        .text
```

```
0000105c
0000105c  00000004  00000000  45505954  45505942  81818181  81818181  81818181
00001078  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001094  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010CC  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

SCREENSHOT - TEST 4



```
R0        :00001067
R1        :0000106b
R2        :00000002
R3        :00000002
R4        :00000000
R5        :00000004
R6        :0000105c
R7        :00001060
R8        :ffffffff
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00001064
R15(pc):00011400
-------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :1
CPU Mode    :System
-------------------
0x000000ff
```

```
                        .data

                            LENGTH:
0000105C:00000004                   .word 4
                            GREATER:
00001060:00000000                   .word 0

                        @ At the time of running, one of the TEST portion below should be unc
                        @ If multiple such TEST portions are uncommented then last one in ord
                        @ effect.

                        @TEST 1 : Here both strings are same, so GREATER will be set to 0.
                        @    STRING1:
                        @            .ascii "BYTE"
                        @    STRING2:
                        @            .ascii "BYTE"

                        @TEST 2 : Here as T > B, GREATER will be set to -1.
                        @    STRING1:
                        @            .ascii "BYTE"
                        @    STRING2:
                        @            .ascii "TYPE"

                        @TEST 3 : Here as B < T, GREATER will be set to 0.
                        @    STRING1:
                        @            .ascii "TYPE"
                        @    STRING2:
                        @            .ascii "BYPE"

                        @TEST 4 : Here, as P < T, GREATER will be set as -1.
                            STRING1:
00001064:45505954                   .ascii "TYPE"
                            STRING2:
00001068:45545954                   .ascii "TYTE"

                        .text
                        .align 2
                        .global _MAIN
```

```
0000105c
0000105C  00000004  FFFFFFFF  45505954  45545954  81818181  106B8181  81818181
00001078  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001094  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010CC  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

EXERCISE – GIVEN TWO STRINGS, CHECK WHETHER THE SECOND STRING IS A SUBSTRING OF THE FIRST ONE.

Given two strings, check whether the second string is a substring of the first one. The starting addresses of two strings are defined by the STRING and SUBSTR variables, respectively. If the string defined by SUBSTR is not present in the string defined by STRING, clear the PRESENT variable; otherwise set the variable with the position of the first occurrence of the second string in the first string.

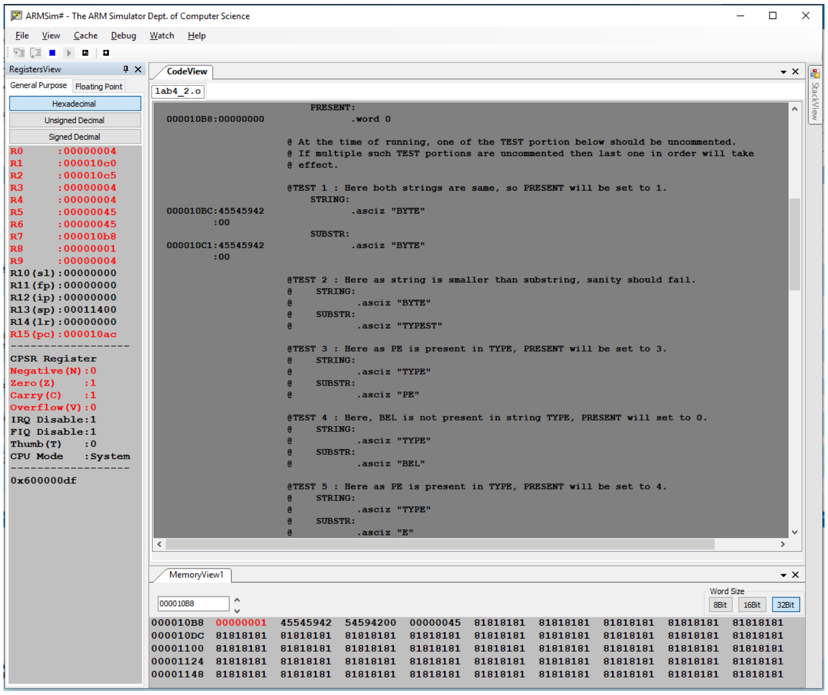
The logic to solve this exercise is hand-coded in 32 bit ARM Assembly and verified on ARMSim Simulator.

The logic used is mentioned in the code file itself and all the necessary instructions are supplied with comments.

Various unit tests are also part of the code file itself. Need to uncomment and run.

All the unit tests passed (various values of STRINGs & SUBSTRINGs).

SCREENSHOT – TEST 1



```
R0        :00000004
R1        :000010c0
R2        :000010c5
R3        :00000004
R4        :00000004
R5        :00000045
R6        :00000045
R7        :000010b8
R8        :00000001
R9        :00000004
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):000010ac
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

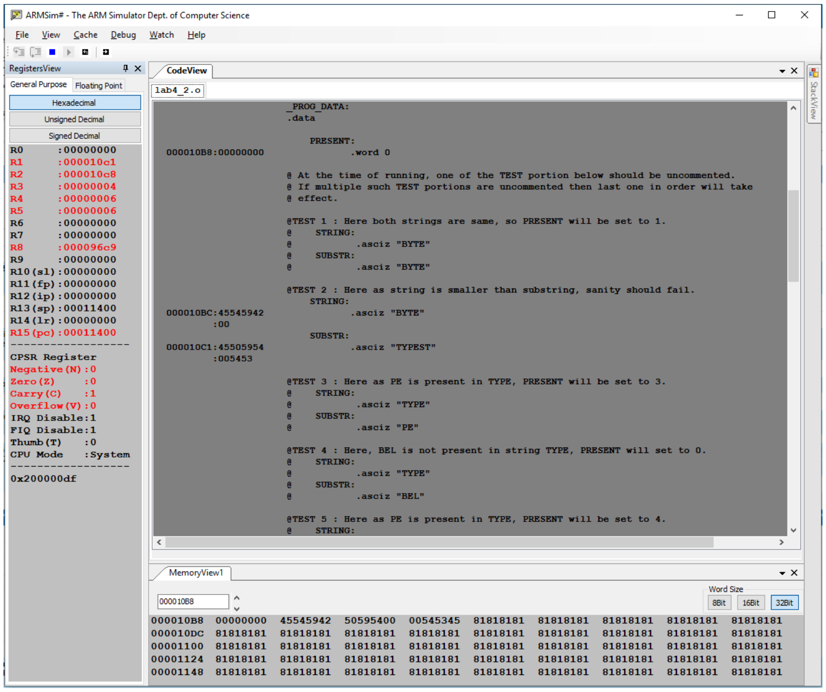
```
                              PRESENT:
000010B8:00000000                      .word 0

                         @ At the time of running, one of the TEST portion below should be uncommented.
                         @ If multiple such TEST portions are uncommented then last one in order will take
                         @ effect.

                         @TEST 1 : Here both strings are same, so PRESENT will be set to 1.
                              STRING:
000010BC:45545942                      .asciz "BYTE"
        :00
                              SUBSTR:
000010C1:45545942                      .asciz "BYTE"
        :00

                         @TEST 2 : Here as string is smaller than substring, sanity should fail.
                         @    STRING:
                         @             .asciz "BYTE"
                         @    SUBSTR:
                         @             .asciz "TYPEST"

                         @TEST 3 : Here as PE is present in TYPE, PRESENT will be set to 3.
                         @    STRING:
                         @             .asciz "TYPE"
                         @    SUBSTR:
                         @             .asciz "PE"

                         @TEST 4 : Here, BEL is not present in string TYPE, PRESENT will set to 0.
                         @    STRING:
                         @             .asciz "TYPE"
                         @    SUBSTR:
                         @             .asciz "BEL"

                         @TEST 5 : Here as PE is present in TYPE, PRESENT will be set to 4.
                         @    STRING:
                         @             .asciz "TYPE"
                         @    SUBSTR:
                         @             .asciz "E"
```

000010B8

Word Size: 8Bit 16Bit 32Bit

```
000010B8 00000001 45545942 54594200 00000045 81818181 81818181 81818181 81818181 81818181
000010DC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001100 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001124 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001148 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

SCREENSHOT – TEST 2

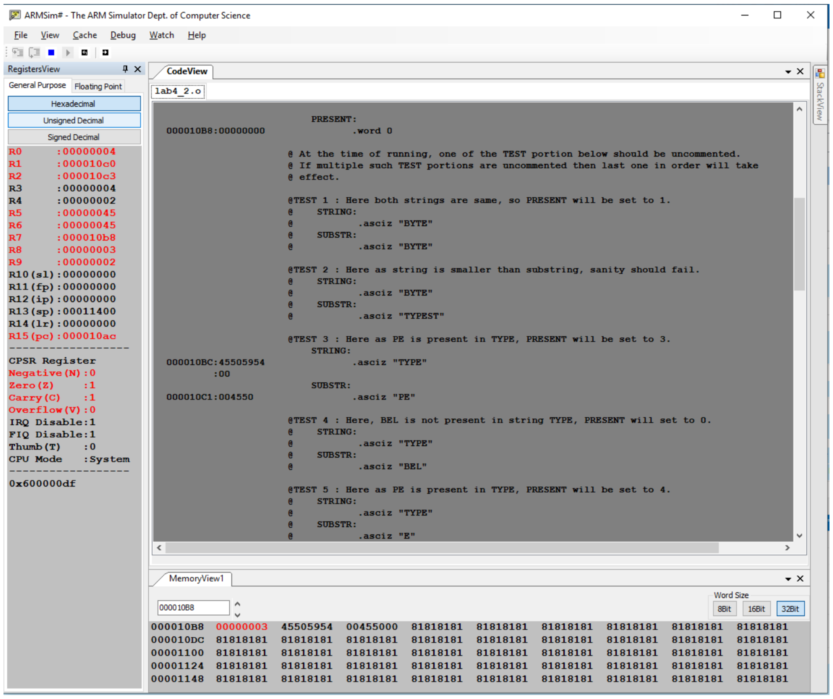


```
R0       :00000000
R1       :000010c1
R2       :000010c8
R3       :00000004
R4       :00000006
R5       :00000006
R6       :00000000
R7       :00000000
R8       :000096c9
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x200000df
```

```
                             _PROG_DATA:
                             .data

                                 PRESENT:
000010B8:00000000                         .word 0

                             @ At the time of running, one of the TEST portion below should be uncommented.
                             @ If multiple such TEST portions are uncommented then last one in order will take
                             @ effect.

                             @TEST 1 : Here both strings are same, so PRESENT will be set to 1.
                             @    STRING:
                             @             .asciz "BYTE"
                             @    SUBSTR:
                             @             .asciz "BYTE"

                             @TEST 2 : Here as string is smaller than substring, sanity should fail.
                                 STRING:
000010BC:45545942                         .asciz "BYTE"
        :00

                                 SUBSTR:
000010C1:45505954                         .asciz "TYPEST"
        :005453

                             @TEST 3 : Here as PE is present in TYPE, PRESENT will be set to 3.
                             @    STRING:
                             @             .asciz "TYPE"
                             @    SUBSTR:
                             @             .asciz "PE"

                             @TEST 4 : Here, BEL is not present in string TYPE, PRESENT will set to 0.
                             @    STRING:
                             @             .asciz "TYPE"
                             @    SUBSTR:
                             @             .asciz "BEL"

                             @TEST 5 : Here as PE is present in TYPE, PRESENT will be set to 4.
                             @    STRING:
```

```
000010B8 00000000 45545942 50595400 00545345 81818181 81818181 81818181 81818181 81818181
000010DC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001100 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001124 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001148 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

SCREENSHOT – TEST 3

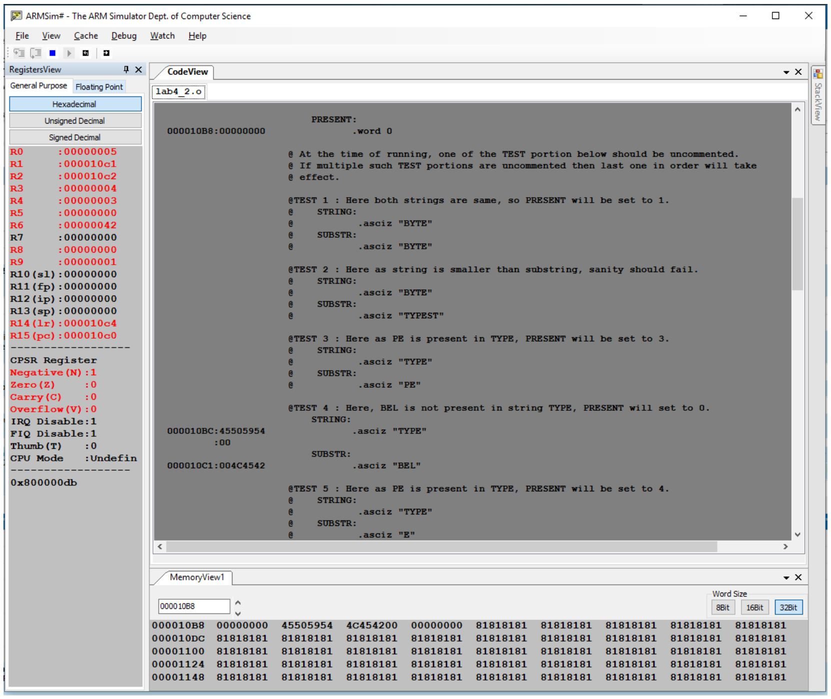


```
R0       :00000004
R1       :000010c0
R2       :000010c3
R3       :00000004
R4       :00000002
R5       :00000045
R6       :00000045
R7       :000010b8
R8       :00000003
R9       :00000002
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):000010ac
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
                                 PRESENT:
000010B8:00000000                         .word 0

                             @ At the time of running, one of the TEST portion below should be uncommented.
                             @ If multiple such TEST portions are uncommented then last one in order will take
                             @ effect.

                             @TEST 1 : Here both strings are same, so PRESENT will be set to 1.
                             @    STRING:
                             @             .asciz "BYTE"
                             @    SUBSTR:
                             @             .asciz "BYTE"

                             @TEST 2 : Here as string is smaller than substring, sanity should fail.
                             @    STRING:
                             @             .asciz "BYTE"
                             @    SUBSTR:
                             @             .asciz "TYPEST"

                             @TEST 3 : Here as PE is present in TYPE, PRESENT will be set to 3.
                                 STRING:
000010BC:45505954                         .asciz "TYPE"
        :00

                                 SUBSTR:
000010C1:004550                           .asciz "PE"

                             @TEST 4 : Here, BEL is not present in string TYPE, PRESENT will set to 0.
                             @    STRING:
                             @             .asciz "TYPE"
                             @    SUBSTR:
                             @             .asciz "BEL"

                             @TEST 5 : Here as PE is present in TYPE, PRESENT will be set to 4.
                             @    STRING:
                             @             .asciz "TYPE"
                             @    SUBSTR:
                             @             .asciz "E"
```

```
000010B8 00000003 45505954 00455000 81818181 81818181 81818181 81818181 81818181 81818181
000010DC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001100 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001124 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001148 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

SCREENSHOT – TEST 4

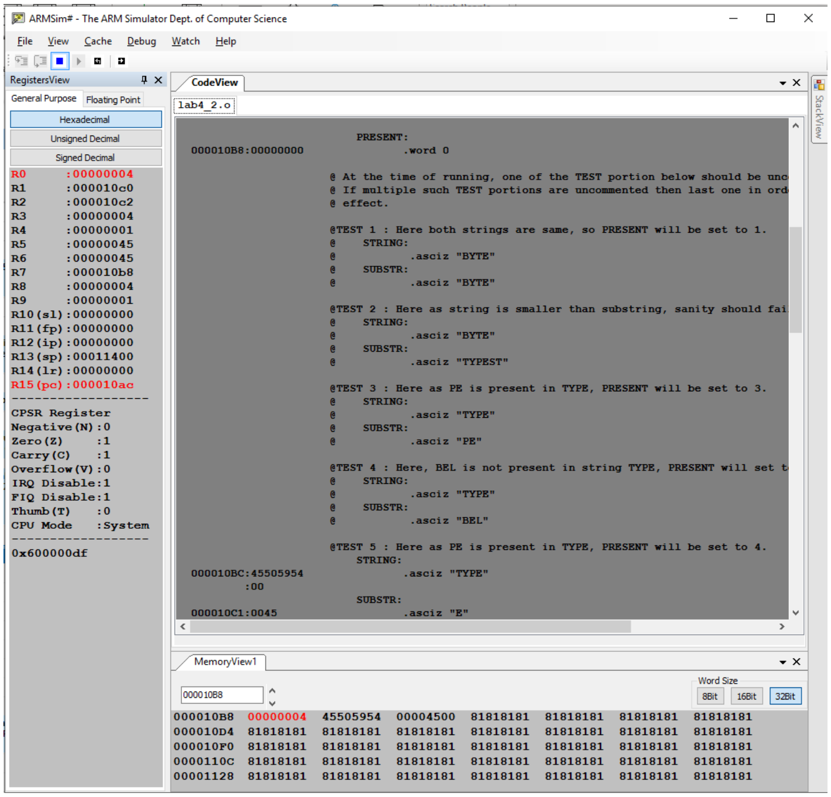


```
R0       :00000005
R1       :000010c1
R2       :000010c2
R3       :00000004
R4       :00000003
R5       :00000000
R6       :00000042
R7       :00000000
R8       :00000000
R9       :00000001
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00000000
R14(lr):000010c4
R15(pc):000010c0
------------------
CPSR Register
Negative(N):1
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :Undefin
------------------
0x800000db
```

```
                          PRESENT:
000010B8:00000000                .word 0

                    @ At the time of running, one of the TEST portion below should be uncommented.
                    @ If multiple such TEST portions are uncommented then last one in order will take
                    @ effect.

                    @TEST 1 : Here both strings are same, so PRESENT will be set to 1.
                    @    STRING:
                    @           .asciz "BYTE"
                    @    SUBSTR:
                    @           .asciz "BYTE"

                    @TEST 2 : Here as string is smaller than substring, sanity should fail.
                    @    STRING:
                    @           .asciz "BYTE"
                    @    SUBSTR:
                    @           .asciz "TYPEST"

                    @TEST 3 : Here as PE is present in TYPE, PRESENT will be set to 3.
                    @    STRING:
                    @           .asciz "TYPE"
                    @    SUBSTR:
                    @           .asciz "PE"

                    @TEST 4 : Here, BEL is not present in string TYPE, PRESENT will set to 0.
                          STRING:
000010BC:45505954               .asciz "TYPE"
        :00

                          SUBSTR:
000010C1:004C4542               .asciz "BEL"

                    @TEST 5 : Here as PE is present in TYPE, PRESENT will be set to 4.
                    @    STRING:
                    @           .asciz "TYPE"
                    @    SUBSTR:
                    @           .asciz "E"
```

```
000010B8
000010B8  00000000  45505954  4C454200  00000000  81818181  81818181  81818181  81818181  81818181
000010DC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001100  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001124  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001148  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

SCREENSHOT – TEST 5



```
R0       :00000004
R1       :000010c0
R2       :000010c2
R3       :00000004
R4       :00000001
R5       :00000045
R6       :00000045
R7       :000010b8
R8       :00000004
R9       :00000001
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
R14(lr):00000000
R15(pc):000010ac
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x600000df
```

```
                          PRESENT:
000010B8:00000000                .word 0

                    @ At the time of running, one of the TEST portion below should be unco
                    @ If multiple such TEST portions are uncommented then last one in ord
                    @ effect.

                    @TEST 1 : Here both strings are same, so PRESENT will be set to 1.
                    @    STRING:
                    @           .asciz "BYTE"
                    @    SUBSTR:
                    @           .asciz "BYTE"

                    @TEST 2 : Here as string is smaller than substring, sanity should fai
                    @    STRING:
                    @           .asciz "BYTE"
                    @    SUBSTR:
                    @           .asciz "TYPEST"

                    @TEST 3 : Here as PE is present in TYPE, PRESENT will be set to 3.
                    @    STRING:
                    @           .asciz "TYPE"
                    @    SUBSTR:
                    @           .asciz "PE"

                    @TEST 4 : Here, BEL is not present in string TYPE, PRESENT will set t
                    @    STRING:
                    @           .asciz "TYPE"
                    @    SUBSTR:
                    @           .asciz "BEL"

                    @TEST 5 : Here as PE is present in TYPE, PRESENT will be set to 4.
                          STRING:
000010BC:45505954               .asciz "TYPE"
        :00

                          SUBSTR:
000010C1:0045                   .asciz "E"
```

```
000010B8
000010B8  00000004  45505954  00004500  81818181  81818181  81818181  81818181
000010D4  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010F0  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000110C  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001128  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```