

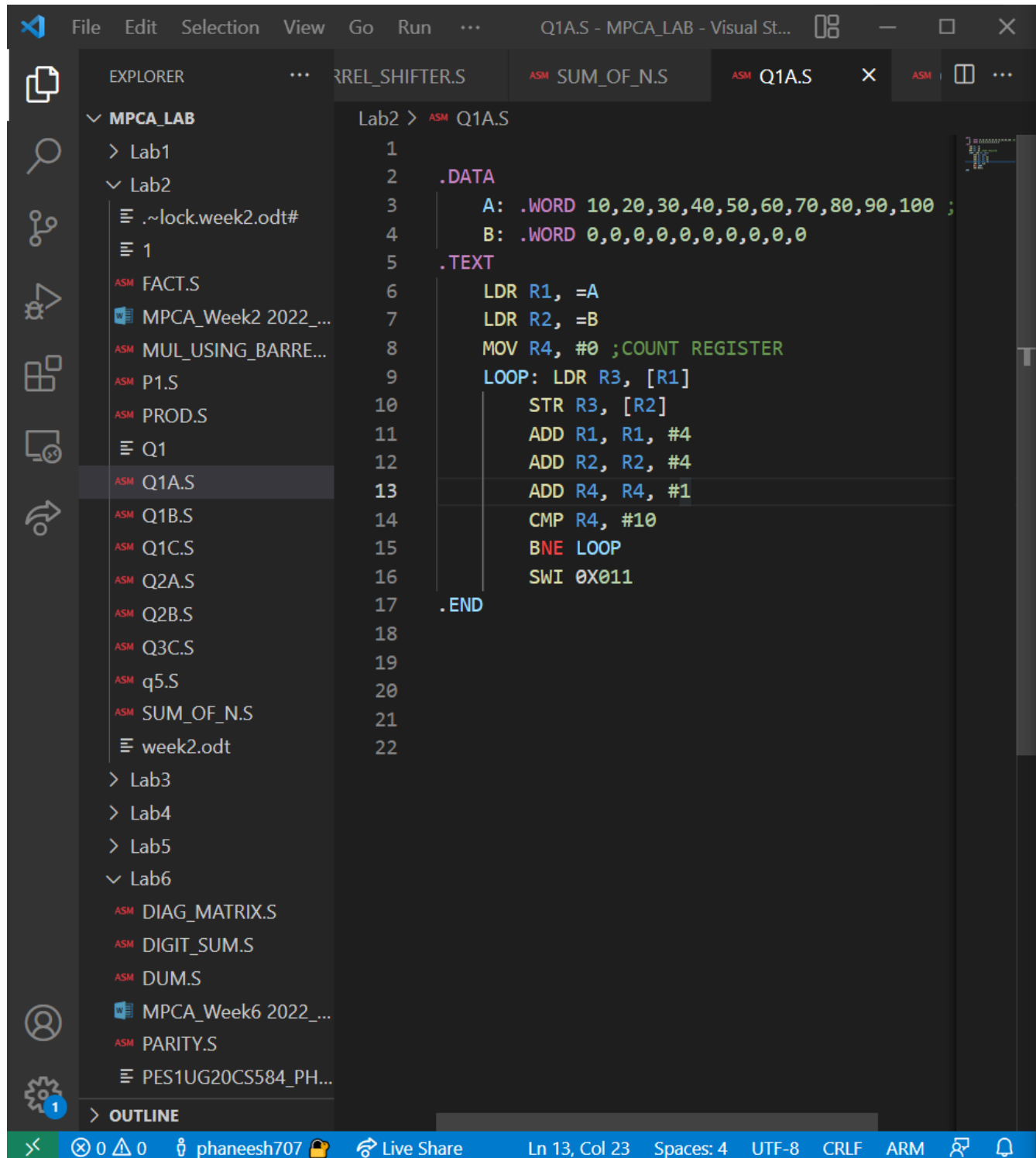
**ARM7TDMI
PROGRAMMING
WEEK-2**

**PHANEESH
PES1UG20CS584
J SECTION**

1. Write a program in ARM7TDMI-ISA to copy a block of N data items from Location A to Location B.

a) Use Full word (.word directive)

CODE:



```
1
2 .DATA
3     A: .WORD 10,20,30,40,50,60,70,80,90,100 ;
4     B: .WORD 0,0,0,0,0,0,0,0,0,0
5 .TEXT
6     LDR R1, =A
7     LDR R2, =B
8     MOV R4, #0 ;COUNT REGISTER
9     LOOP: LDR R3, [R1]
10         STR R3, [R2]
11         ADD R1, R1, #4
12         ADD R2, R2, #4
13         ADD R4, R4, #1
14
15         CMP R4, #10
16         BNE LOOP
17     SWI 0X011
18 .END
19
20
21
22
```

OUTPUT:

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView Q1A.S

General Purpose

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 4188

R2 : 4228

R3 : 100

R4 : 10

R5 : 0

R6 : 0

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4136

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 : Sys

0x600000df

.DATA

00001034: A: .WORD 10,20,30,40,50,60,70,80,90,100 ;WORD

0000105C: B: .WORD 0,0,0,0,0,0,0,0,0,0

.TEXT

00001000:E59F1024 LDR R1, =A

00001004:E59F2024 LDR R2, =B

00001008:E3A04000 MOV R4, #0 ;COUNT REGISTER

0000100C:E5913000 LOOP: LDR R3, [R1]

00001010:E5823000 STR R3, [R2]

00001014:E2811004 ADD R1, R1, #4

00001018:E2822004 ADD R2, R2, #4

0000101C:E2844001 ADD R4, R4, #1

00001020:E354000A CMP R4, #10

00001024:1AFFFFFF8 BNE LOOP

00001028:EF000011 SWI 0X011

0000102C:00001034 .END

00001030:0000105C

MemoryView0

Word Size

8Bit 16Bit 32Bit

0000105C

0000105C 0000000A 00000014 0000001E 00000028 00000032

00001070 0000003C 00000046 00000050 0000005A 00000064

00001084 81818181 81818181 81818181 81818181 81818181

00001098 81818181 81818181 81818181 81818181 81818181

000010AC 81818181 81818181 81818181 81818181 81818181

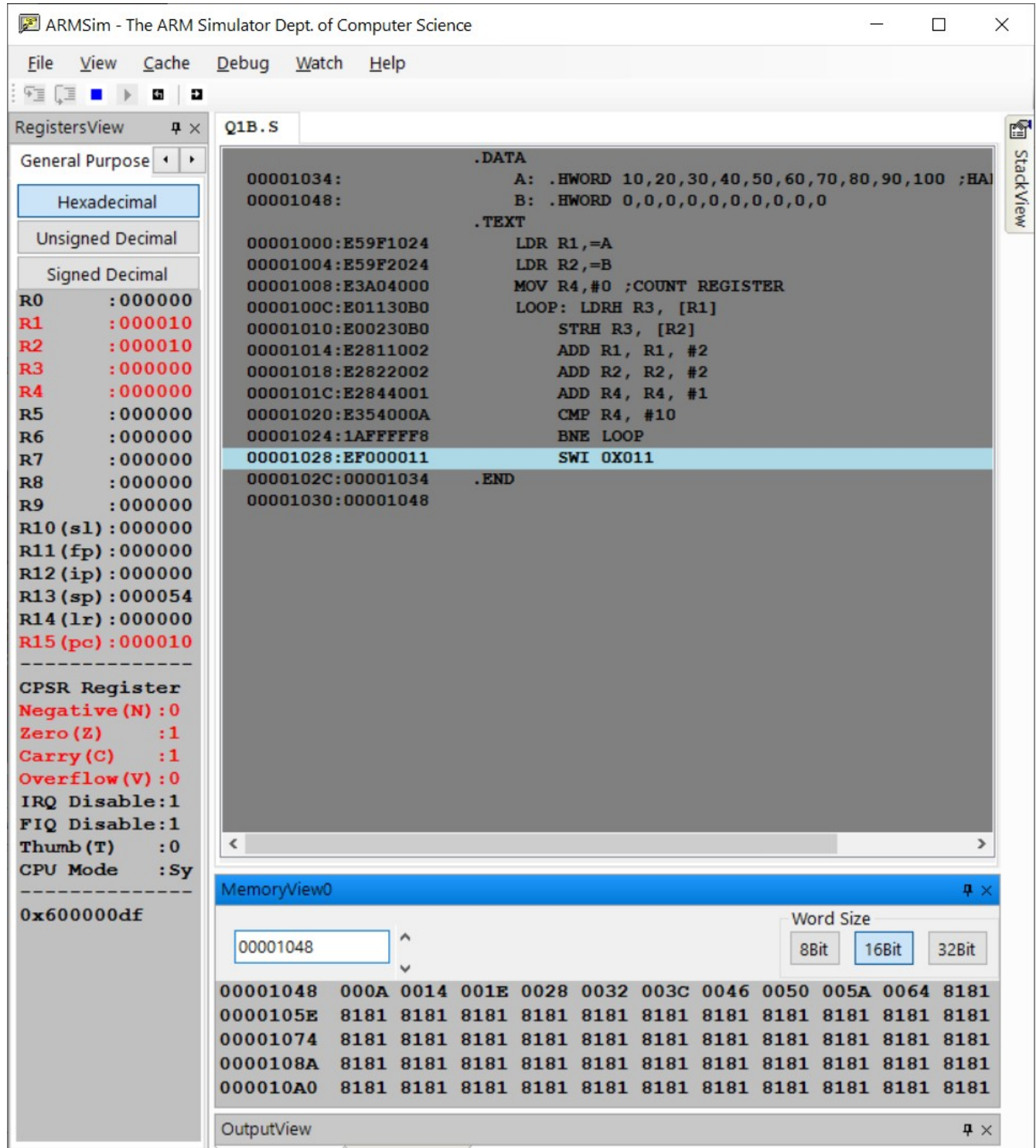
OutputView

b) Use Half word(.Hword directive)

CODE:

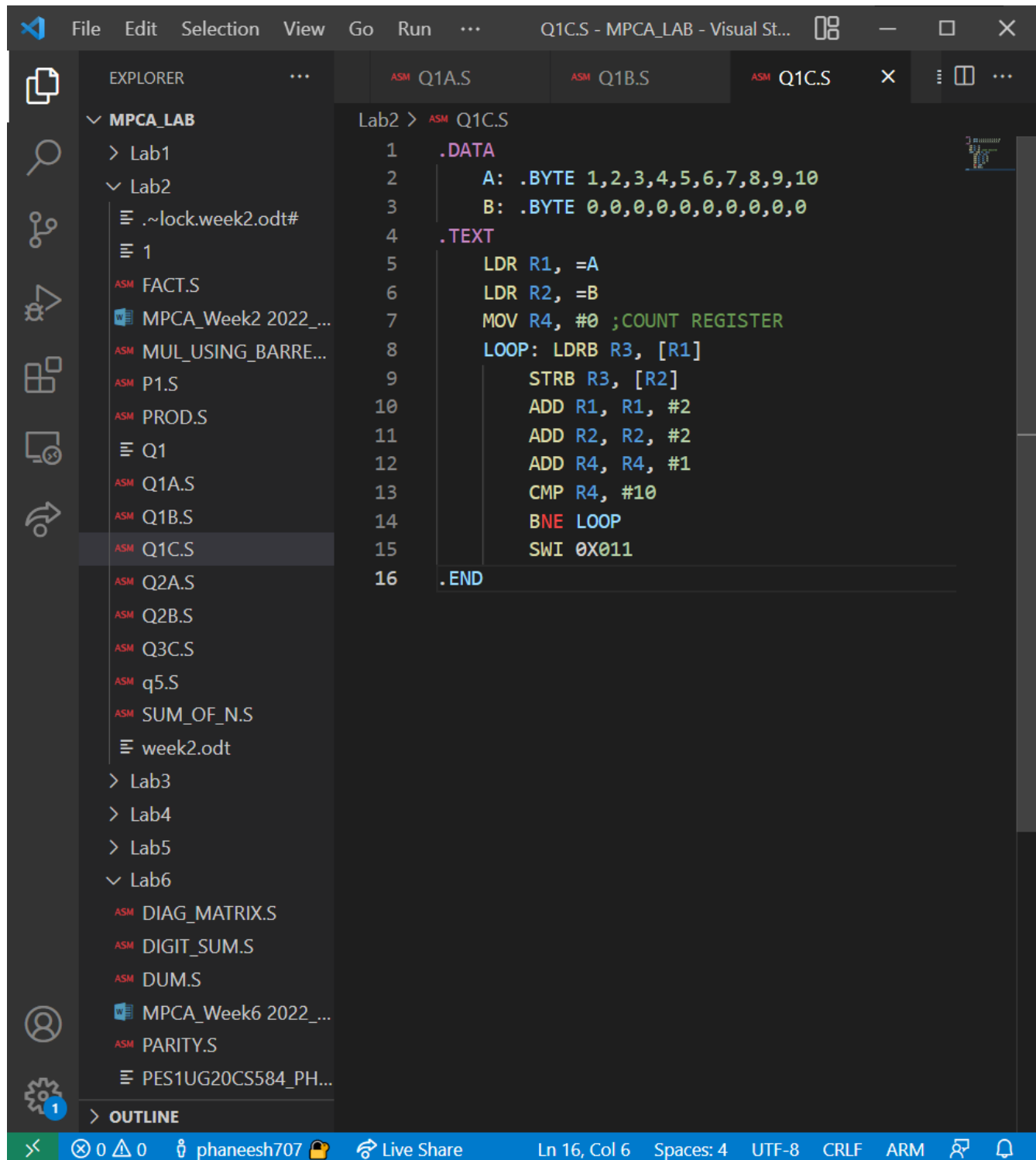
```
1  .DATA
2      A: .HWORD 10,20,30,40,50,60,70,80,90,100
3      B: .HWORD 0,0,0,0,0,0,0,0,0,0
4  .TEXT
5      LDR R1,=A
6      LDR R2,=B
7      MOV R4,#0 ;COUNT REGISTER
8      LOOP: LDRH R3, [R1]
9              STRH R3, [R2]
10             ADD R1, R1, #2
11             ADD R2, R2, #2
12             ADD R4, R4, #1
13             CMP R4, #10
14             BNE LOOP
15             SWI 0X011
16  .END
```

OUTPUT:



c) Use Byte wise (.Byte directive)

CODE:



```
1  .DATA
2      A: .BYTE 1,2,3,4,5,6,7,8,9,10
3      B: .BYTE 0,0,0,0,0,0,0,0,0,0
4  .TEXT
5      LDR R1, =A
6      LDR R2, =B
7      MOV R4, #0 ;COUNT REGISTER
8      LOOP: LDRB R3, [R1]
9              STRB R3, [R2]
10             ADD R1, R1, #2
11             ADD R2, R2, #2
12             ADD R4, R4, #1
13             CMP R4, #10
14             BNE LOOP
15             SWI 0X011
16 .END
```


OUTPUT:

ARMSim - The ARM Simulator Dept. of Computer Science

FileViewCacheDebugWatchHelp

RegistersView

Q1C.S

StackView

General Purpose

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 000000

R1 : 000010

R2 : 000010

R3 : 000000

R4 : 000000

R5 : 000000

R6 : 000000

R7 : 000000

R8 : 000000

R9 : 000000

R10 (s1) : 000000

R11 (fp) : 000000

R12 (ip) : 000000

R13 (sp) : 000054

R14 (lr) : 000000

R15 (pc) : 000010

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 : Sy

0x600000df

.DATA

00001034: A: .BYTE 1,2,3,4,5,6,7,8,9,10

0000103E: B: .BYTE 0,0,0,0,0,0,0,0,0,0

.TEXT

00001000:E59F1024 LDR R1, =A

00001004:E59F2024 LDR R2, =B

00001008:E3A04000 MOV R4, #0 ;COUNT REGISTER

0000100C:E5D13000 LOOP: LDRB R3, [R1]

00001010:E5C23000 STRB R3, [R2]

00001014:E2811002 ADD R1, R1, #2

00001018:E2822002 ADD R2, R2, #2

0000101C:E2844001 ADD R4, R4, #1

00001020:E354000A CMP R4, #10

00001024:1AFFFFF8 BNE LOOP

00001028:EF000011 SWI 0X011

0000102C:00001034 .END

00001030:0000103E

MemoryView0

Word Size

8Bit

16Bit

32Bit

00001060

00001060 81818181 81818181 81818181 81818181 81818181

00001074 81818181 81818181 81818181 81818181 81818181

00001088 81818181 81818181 81818181 81818181 81818181

0000109C 81818181 81818181 81818181 81818181 81818181

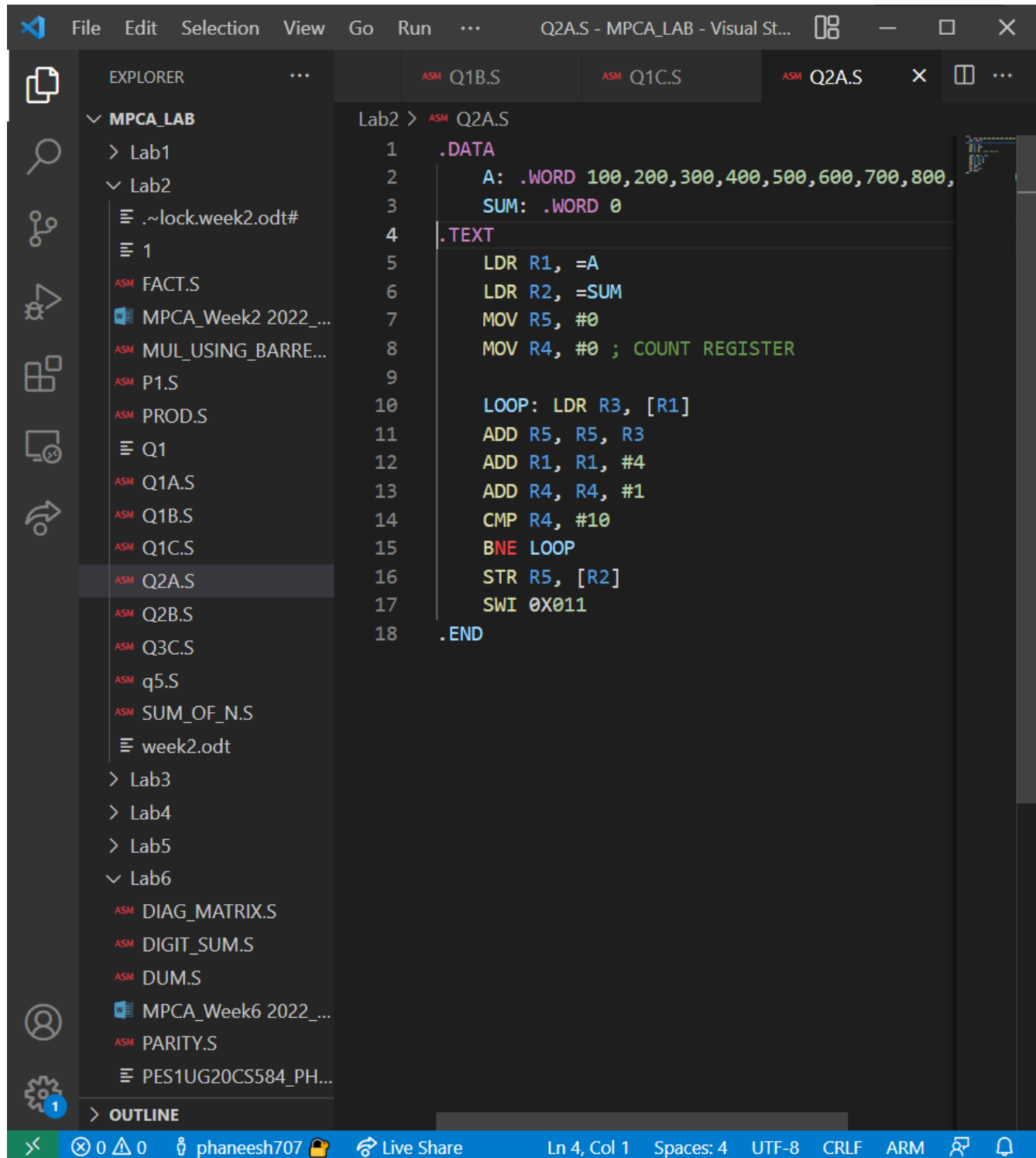
000010B0 81818181 81818181 81818181 81818181 81818181

OutputView

2. Write a program in ARM7TDMI-ISA to find the sum of N data items in the memory. Store the result in the memory location.

a) Use Full word (.word directive)

CODE:



```
Lab2 > ASM Q2A.S
1  .DATA
2      A: .WORD 100,200,300,400,500,600,700,800,
3      SUM: .WORD 0
4  .TEXT
5      LDR R1, =A
6      LDR R2, =SUM
7      MOV R5, #0
8      MOV R4, #0 ; COUNT REGISTER
9
10     LOOP: LDR R3, [R1]
11     ADD R5, R5, R3
12     ADD R1, R1, #4
13     ADD R4, R4, #1
14     CMP R4, #10
15     BNE LOOP
16     STR R5, [R2]
17     SWI 0X011
18 .END
```


OUTPUT:

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView Q2A.S

General Purpose

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 :000000
R1 :000000
R2 :000000
R3 :000000
R4 :000000
R5 :000000
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10 (s1):000000
R11 (fp):000000
R12 (ip):000000
R13 (sp):000054
R14 (lr):000000
R15 (pc):000010

CPSR Register
Negative (N) :0
Zero (Z) :0
Carry (C) :0
Overflow (V) :0
IRQ Disable:1
FIQ Disable:1
Thumb (T) :0
CPU Mode :Sy

0x000000df

.DATA
00001038: A: .WORD 100,200,300,400,500,600,700,800,900
00001060: SUM: .WORD 0

.TEXT
00001000:E59F1028 LDR R1, =A
00001004:E59F2028 LDR R2, =SUM
00001008:E3A05000 MOV R5, #0
0000100C:E3A04000 MOV R4, #0 ; COUNT REGISTER

00001010:E5913000 LOOP: LDR R3, [R1]
00001014:E0855003 ADD R5, R5, R3
00001018:E2811004 ADD R1, R1, #4
0000101C:E2844001 ADD R4, R4, #1
00001020:E354000A CMP R4, #10
00001024:1AFFFFF9 BNE LOOP
00001028:E5825000 STR R5, [R2]
0000102C:EF000011 SWI 0X011
00001030:00001038 .END
00001034:00001060

MemoryView0

Word Size
8Bit 16Bit 32Bit

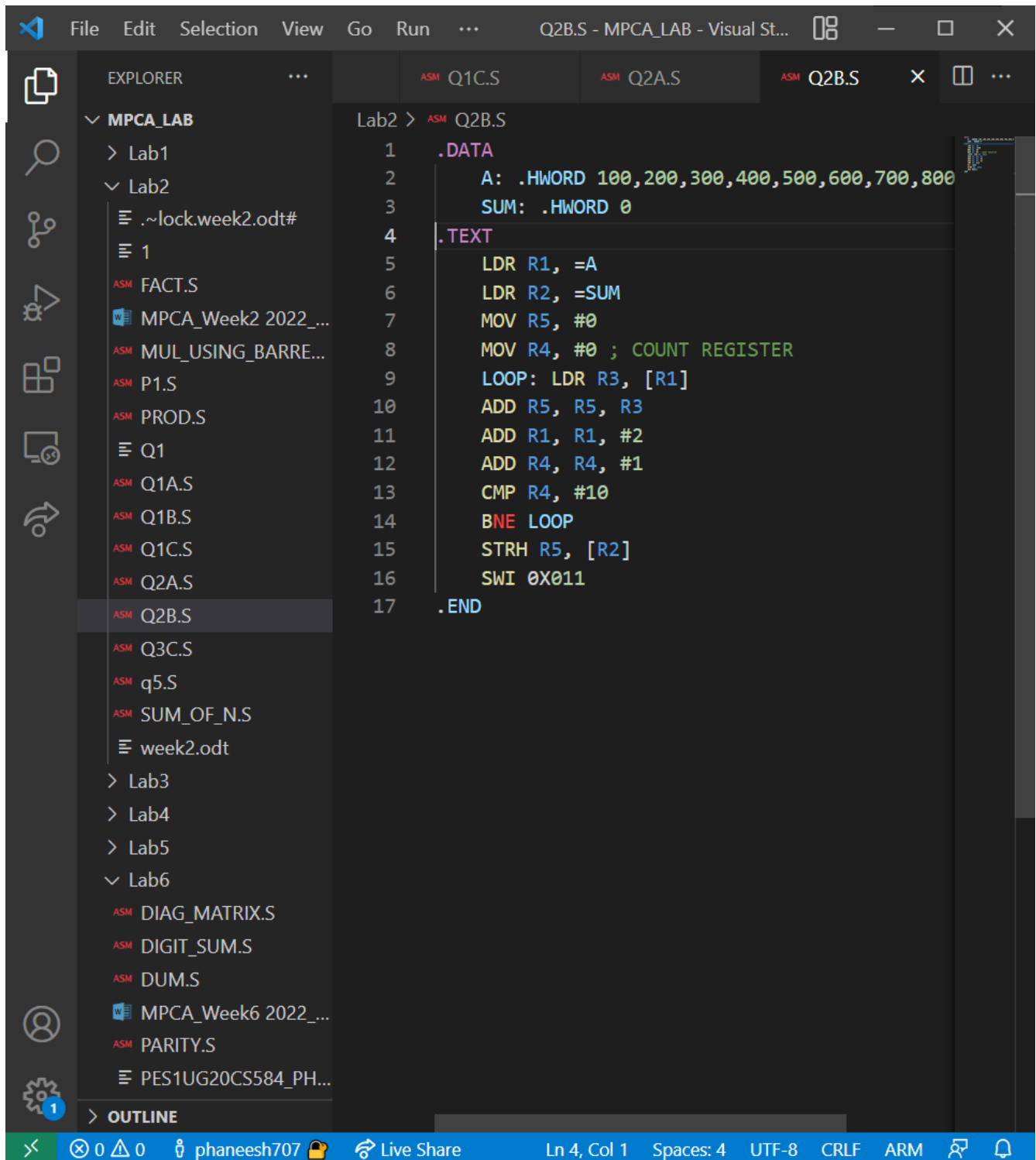
0000104C

0000104C 0258 0000 02BC 0000 0320 0000 0384 0000 03E8 0000 0000
00001062 0000 8181 8181 8181 8181 8181 8181 8181 8181 8181
00001078 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
0000108E 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181
000010A4 8181 8181 8181 8181 8181 8181 8181 8181 8181 8181

OutputView

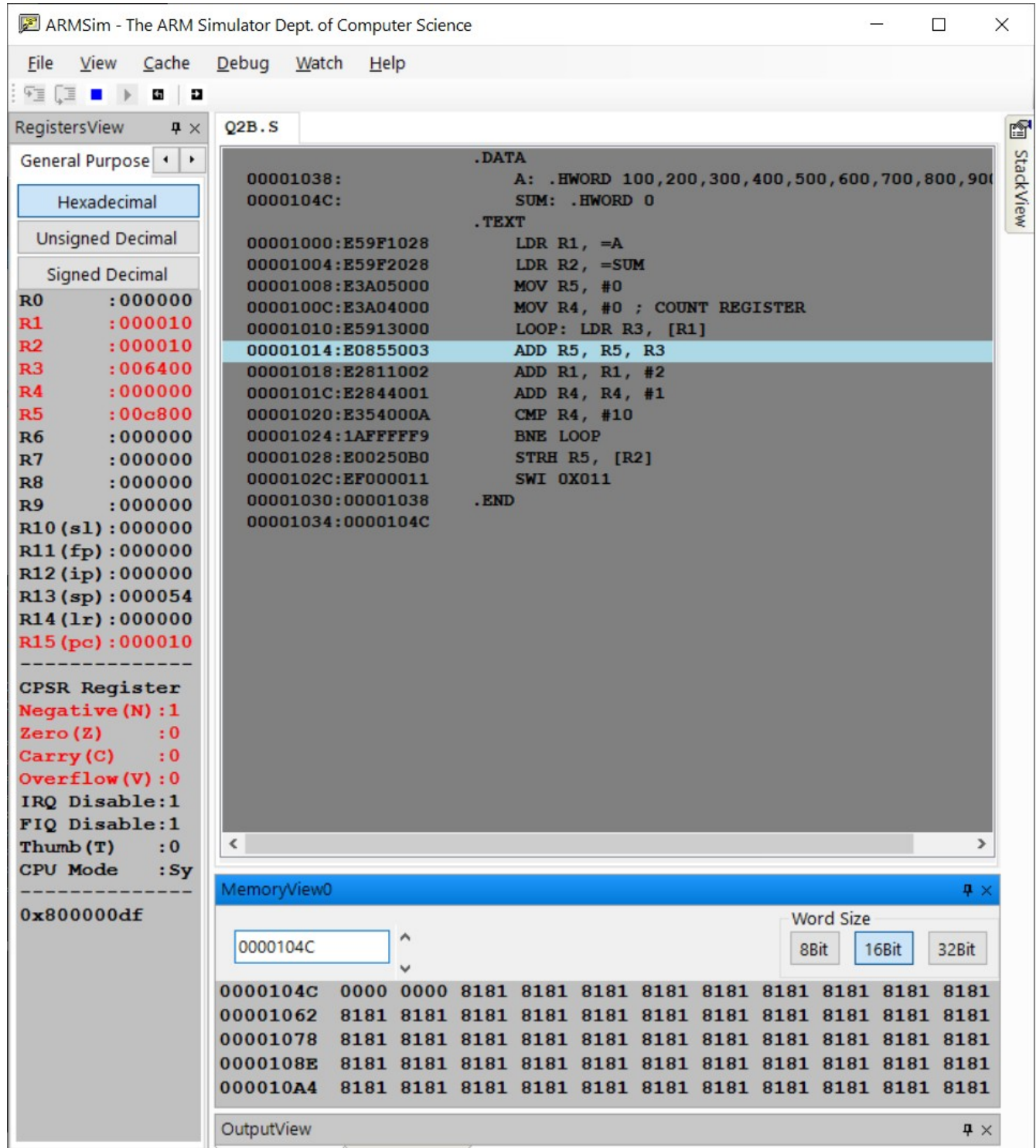
b) Use Half word(.Hword directive)

CODE:



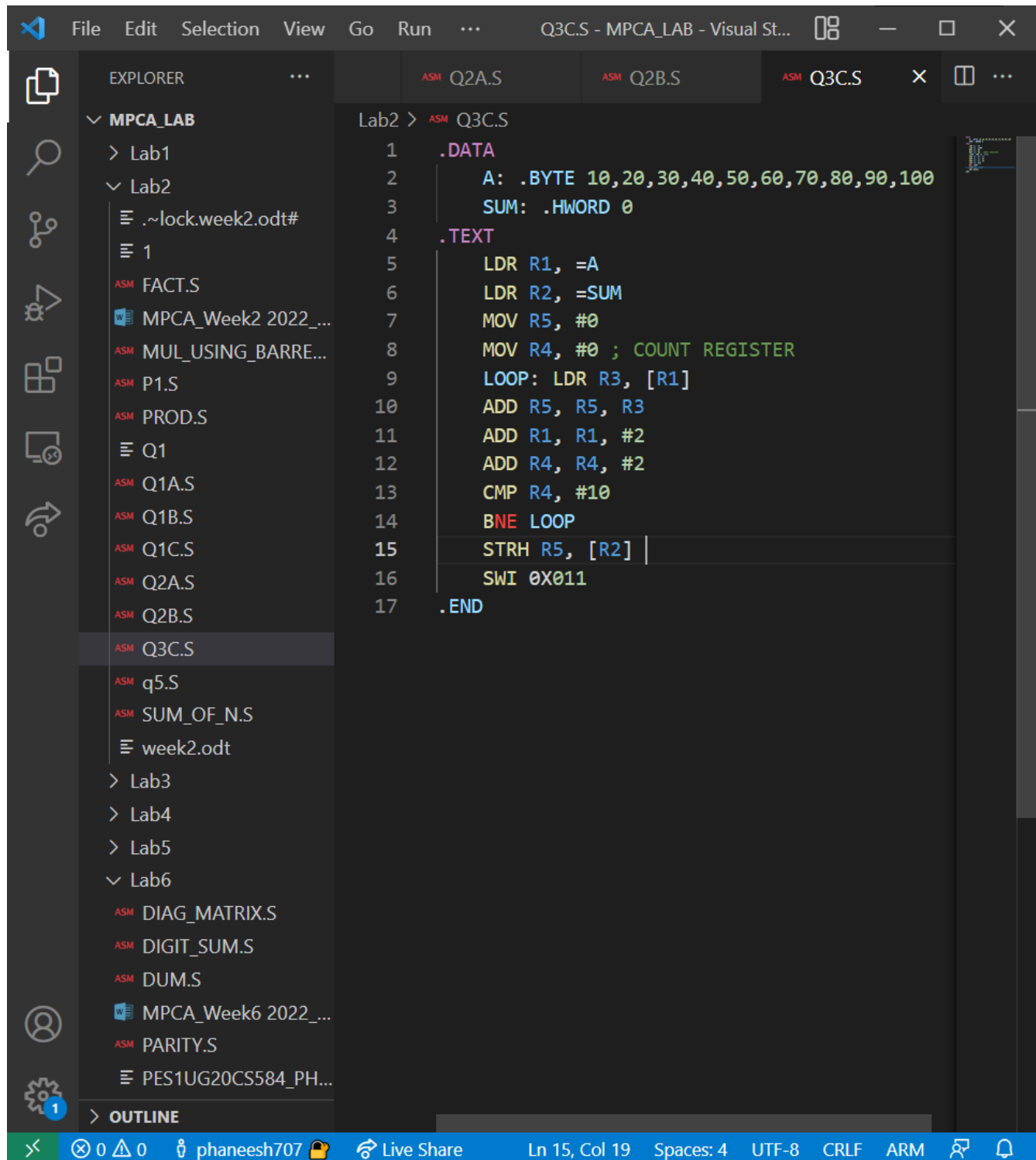
```
1  .DATA
2      A: .HWORD 100,200,300,400,500,600,700,800
3      SUM: .HWORD 0
4  .TEXT
5      LDR R1, =A
6      LDR R2, =SUM
7      MOV R5, #0
8      MOV R4, #0 ; COUNT REGISTER
9      LOOP: LDR R3, [R1]
10     ADD R5, R5, R3
11     ADD R1, R1, #2
12     ADD R4, R4, #1
13     CMP R4, #10
14     BNE LOOP
15     STRH R5, [R2]
16     SWI 0X011
17 .END
```

OUTPUT:



c) Use Byte wise (.Byte directive)

CODE:



```
Q3C.S - MPCA_LAB - Visual St...
File Edit Selection View Go Run ...
EXPLORER
MPCA_LAB
  Lab1
  Lab2
    .~lock.week2.odt#
    1
    FACT.S
    MPCA_Week2 2022_...
    MUL_USING_BARRE...
    P1.S
    PROD.S
    Q1
    Q1A.S
    Q1B.S
    Q1C.S
    Q2A.S
    Q2B.S
    Q3C.S
    q5.S
    SUM_OF_N.S
    week2.odt
  Lab3
  Lab4
  Lab5
  Lab6
    DIAG_MATRIX.S
    DIGIT_SUM.S
    DUM.S
    MPCA_Week6 2022_...
    PARITY.S
    PES1UG20CS584_PH...
  OUTLINE

Lab2 > Q3C.S
1 .DATA
2   A: .BYTE 10,20,30,40,50,60,70,80,90,100
3   SUM: .HWORD 0
4 .TEXT
5   LDR R1, =A
6   LDR R2, =SUM
7   MOV R5, #0
8   MOV R4, #0 ; COUNT REGISTER
9   LOOP: LDR R3, [R1]
10  ADD R5, R5, R3
11  ADD R1, R1, #2
12  ADD R4, R4, #2
13  CMP R4, #10
14  BNE LOOP
15  STRH R5, [R2]
16  SWI 0X011
17 .END

Ln 15, Col 19 Spaces: 4 UTF-8 CRLF ARM
```

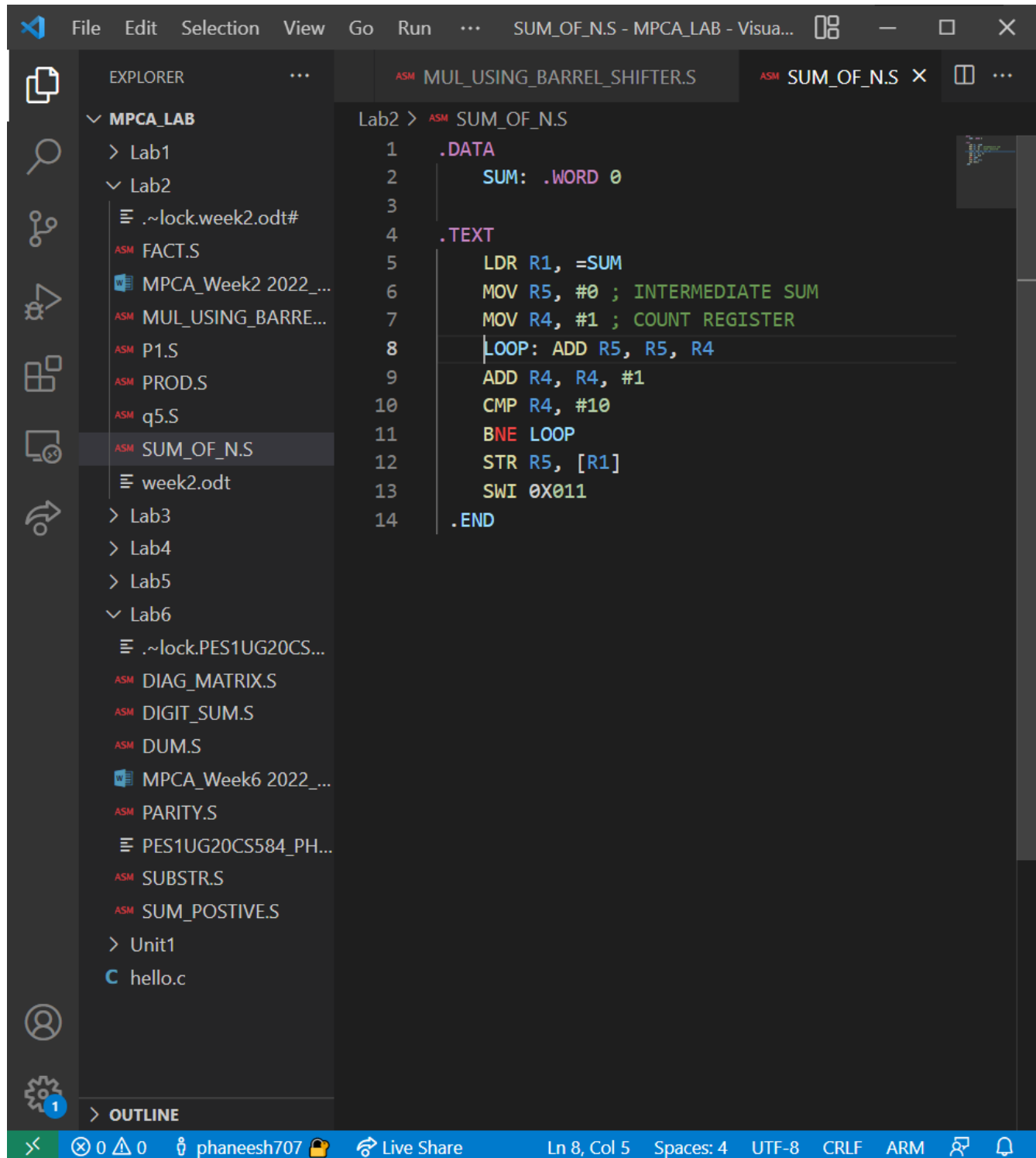

OUTPUT:

The screenshot displays the ARMSim - The ARM Simulator interface, which is a software tool for simulating ARM processors. The interface is divided into several panels:

- RegistersView:** This panel on the left shows the state of the ARM registers. It includes a "General Purpose" section with a dropdown menu set to "Hexadecimal". Below this, the registers R0 through R15 are listed with their current values. R15 (PC) is highlighted in red. The CPSR Register is also shown with various flags like Negative (N), Zero (Z), Carry (C), and Overflow (V) set to 0, and IRQ/FIQ Disable flags set to 1. The CPU Mode is shown as "Sy" (System).
- MemoryView:** This panel in the middle shows the memory contents. It has a "Word Size" dropdown set to "16Bit". The memory address 00001048 is selected, and the contents of the memory are displayed in a table format, showing values like 8181, 8181, etc.
- OutputView:** This panel at the bottom is currently empty, showing the output of the simulation.
- Assembly View:** The main area on the right displays the assembly code being executed. It includes sections for .DATA, .TEXT, and .END. The code defines a loop that increments R5 by R3, then increments R1 and R4 by 2, and checks if R4 is greater than or equal to 10. If so, it branches back to the start of the loop.

3. Write a program in ARM7TDMI-ISA to find the sum of N natural numbers. Store the result in the memory location.

CODE:



The screenshot shows the Visual Studio Code interface with a project named 'MPCA_LAB'. The Explorer sidebar on the left lists various files, including 'SUM_OF_N.S' which is currently selected. The main editor window displays the assembly code for 'SUM_OF_N.S'. The code is written in ARM7TDMI-ISA and includes a data section for a memory location 'SUM' and a text section containing a loop to calculate the sum of the first 10 natural numbers. The status bar at the bottom indicates the current position is Line 8, Column 5.

```
1  .DATA
2      SUM: .WORD 0
3
4  .TEXT
5      LDR R1, =SUM
6      MOV R5, #0 ; INTERMEDIATE SUM
7      MOV R4, #1 ; COUNT REGISTER
8  LOOP: ADD R5, R5, R4
9      ADD R4, R4, #1
10     CMP R4, #10
11     BNE LOOP
12     STR R5, [R1]
13     SWI 0X011
14 .END
```


OUTPUT:

ARMSim - The ARM Simulator Dept. of Computer Science

FileViewCacheDebugWatchHelp

RegistersView

General Purpose

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 4136

R2 : 0

R3 : 0

R4 : 10

R5 : 45

R6 : 0

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4128

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 : Sy

0x600000df

SUM_OF_N.S

.DATA

00001028: SUM: .WORD 0

.TEXT

00001000:E59F101C LDR R1, =SUM

00001004:E3A05000 MOV R5, #0 ; INTERMEDIATE SUM

00001008:E3A04001 MOV R4, #1 ; COUNT REGISTER

0000100C:E0855004 LOOP: ADD R5, R5, R4

00001010:E2844001 ADD R4, R4, #1

00001014:E354000A CMP R4, #10

00001018:1AFFFFFB BNE LOOP

0000101C:E5815000 STR R5, [R1]

00001020:EF000011 SWI 0X011

00001024:00001028 .END

StackView

MemoryView0

0000102C

Word Size

8Bit16Bit32Bit

0000102C 81818181 81818181 81818181 81818181 81818181

00001040 81818181 81818181 81818181 81818181 81818181

00001054 81818181 81818181 81818181 81818181 81818181

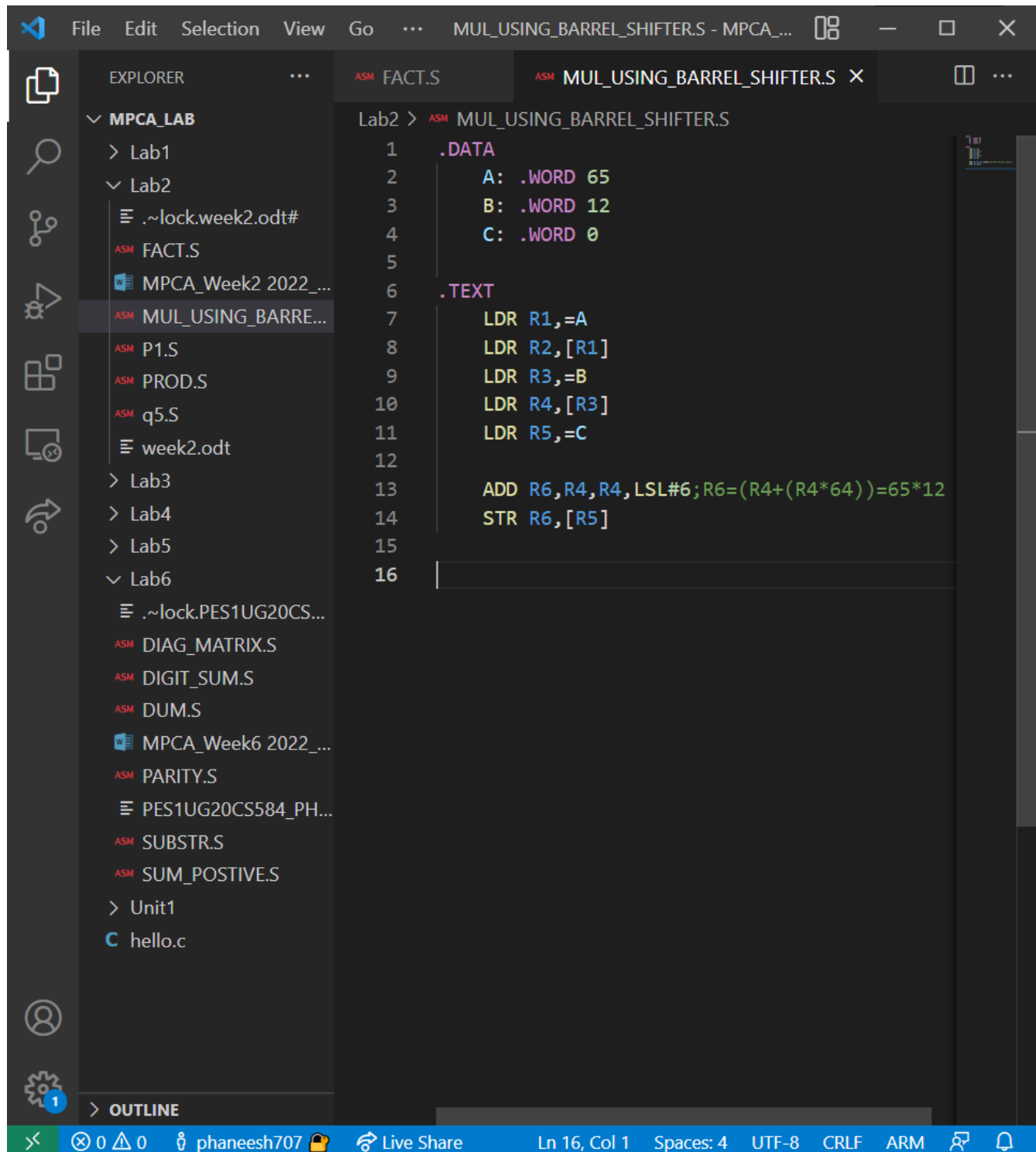
00001068 81818181 81818181 81818181 81818181 81818181

0000107C 81818181 81818181 81818181 81818181 81818181

OutputView

4. Write a program in ARM7TDMI-ISA to find the product of two 32-bit numbers using barrel shifter.

CODE:



The screenshot shows a code editor with a dark theme. The Explorer panel on the left shows a project structure with folders 'Lab1' through 'Lab6' and various files. The active file is 'MUL_USING_BARREL_SHIFTER.S'. The main editor area displays the following assembly code:

```
1  .DATA
2      A: .WORD 65
3      B: .WORD 12
4      C: .WORD 0
5
6  .TEXT
7      LDR R1,=A
8      LDR R2,[R1]
9      LDR R3,=B
10     LDR R4,[R3]
11     LDR R5,=C
12
13     ADD R6,R4,R4,LSL#6;R6=(R4+(R4*64))=65*12
14     STR R6,[R5]
15
16
```

The status bar at the bottom indicates the current position is Ln 16, Col 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and the ARM architecture selected.

OUTPUT :

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0
R1 : 4136
R2 : 65
R3 : 4140
R4 : 12
R5 : 4144
R6 : 780
R7 : 0
R8 : 0
R9 : 0
R10 (s1) : 0
R11 (fp) : 0
R12 (ip) : 0
R13 (sp) : 21504
R14 (lr) : 0
R15 (pc) : 4124

CPSR Register
Negative (N) : 0
Zero (Z) : 0
Carry (C) : 0
Overflow (V) : 0
IRQ Disable : 1
FIQ Disable : 1
Thumb (T) : 0
CPU Mode : Sy

0x000000df

MUL_USING_BARREL_SHIFTER.S

.DATA
00001028: A: .WORD 65
0000102C: B: .WORD 12
00001030: C: .WORD 0

.TEXT
00001000:E59F1014 LDR R1,=A
00001004:E5912000 LDR R2,[R1]
00001008:E59F3010 LDR R3,=B
0000100C:E5934000 LDR R4,[R3]
00001010:E59F500C LDR R5,=C
00001014:E0846304 ADD R6,R4,R4,LSL#6;R6=(R4+(R4*64))=65*12
00001018:E5856000 STR R6,[R5]

MemoryView0

Word Size
8Bit 16Bit 32Bit

0000102C 0000000C 0000030C 81818181 81818181 81818181
00001040 81818181 81818181 81818181 81818181 81818181
00001054 81818181 81818181 81818181 81818181 81818181
00001068 81818181 81818181 81818181 81818181 81818181
0000107C 81818181 81818181 81818181 81818181 81818181

OutputView

5. Convert the following statement in C language into an ALP using ARM7TDMI - ISA.

```
IF ([A]==[B]) then C=[A]+[B];  
    ELSE IF ([B]==[C]) D=[A]-[B];  
    ELSE E=[A]*[B]
```

CODE:



```
Lab2 > q5.S  
1  .DATA  
2      A: .WORD 200  
3      B: .WORD 200  
4      C: .WORD 300  
5      D: .WORD 0  
6      E: .WORD 0  
7  
8  .TEXT  
9      LDR R1,=A  
10     LDR R2,=B  
11     LDR R3,=C  
12     LDR R4,=D  
13     LDR R5,=E  
14  
15     LDR R6,[R1]  
16     LDR R7,[R2]  
17     LDR R8,[R3]  
18     LDR R9,[R4]  
19     LDR R10,[R5]  
20  
21     CMP R6,R7  
22     BEQ LOOP1  
23     CMP R7,R8  
24     BEQ LOOP2  
25     MUL R10,R6,R7  
26     SWI 0X011  
27  
28     LOOP1: ADD R8,R7,R6  
29             SWI 0X011  
30  
31     LOOP2: SUB R9,R7,R6  
32             SWI 0X011  
33  
34
```

OUTPUT :

1.WHEN A = B

RegistersView

General Purpose Float

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 4208

R2 : 4212

R3 : 4216

R4 : 4220

R5 : 4224

R6 : 200

R7 : 200

R8 : 400

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4172

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

q5.S

00001078: C: .WORD 300

0000107C: D: .WORD 0

00001080: E: .WORD 0

.TEXT

00001000:E59F1054 LDR R1,=A

00001004:E59F2054 LDR R2,=B

00001008:E59F3054 LDR R3,=C

0000100C:E59F4054 LDR R4,=D

00001010:E3A05D42 LDR R5,=E

00001014:E5916000 LDR R6,[R1]

00001018:E5927000 LDR R7,[R2]

0000101C:E5938000 LDR R8,[R3]

00001020:E5949000 LDR R9,[R4]

00001024:E595A000 LDR R10,[R5]

00001028:E1560007 CMP R6,R7

0000102C:0A000004 BEQ LOOP1

00001030:E1570008 CMP R7,R8

00001034:0A000005 BEQ LOOP2

00001038:E00A0796 MUL R10,R6,R7

0000103C:E585A000 STR R10,[R5]

00001040:EF000011 SWI 0X011

00001044:E0878006 LOOP1: ADD R8,R7,R6

00001048:E5838000 STR R8,[R3]

0000104C:EF000011 SWI 0X011

00001050:E0479006 LOOP2: SUB R9,R7,R6

00001054:E5849000 STR R9,[R4]

MemoryView0

Word Size

8Bit

16Bit

32Bit

00001078

00001078 00000190 00000000 00000000 81818181

00001088 81818181 81818181 81818181 81818181

00001098 81818181 81818181 81818181 81818181

000010A8 81818181 81818181 81818181 81818181

000010B8 81818181 81818181 81818181 81818181

000010C8 81818181 81818181 81818181 81818181

2. WHEN B=C

RegistersView

General Purpose Float

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 4208

R2 : 4212

R3 : 4216

R4 : 4220

R5 : 4224

R6 : 500

R7 : 200

R8 : 200

R9 : 300

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4184

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

q5.S

.TEXT

00001000:E59F1054 LDR R1,=A

00001004:E59F2054 LDR R2,=B

00001008:E59F3054 LDR R3,=C

0000100C:E59F4054 LDR R4,=D

00001010:E3A05D42 LDR R5,=E

00001014:E5916000 LDR R6,[R1]

00001018:E5927000 LDR R7,[R2]

0000101C:E5938000 LDR R8,[R3]

00001020:E5949000 LDR R9,[R4]

00001024:E595A000 LDR R10,[R5]

00001028:E1560007 CMP R6,R7

0000102C:0A000004 BEQ LOOP1

00001030:E1570008 CMP R7,R8

00001034:0A000005 BEQ LOOP2

00001038:E00A0796 MUL R10,R6,R7

0000103C:E585A000 STR R10,[R5]

00001040:EF000011 SWI 0X011

00001044:E0878006 LOOP1: ADD R8,R7,R6

00001048:E5838000 STR R8,[R3]

0000104C:EF000011 SWI 0X011

00001050:E0469007 LOOP2: SUB R9,R6,R7

00001054:E5849000 STR R9,[R4]

00001058:EF000011 SWI 0X011

MemoryView0

Word Size

8Bit 16Bit 32Bit

0000107C 0000012C 00000000 81818181 81818181

0000108C 81818181 81818181 81818181 81818181

0000109C 81818181 81818181 81818181 81818181

000010AC 81818181 81818181 81818181 81818181

000010BC 81818181 81818181 81818181 81818181

000010CC 81818181 81818181 81818181 81818181

3. IN ALL OTHER CASES

RegistersView

General Purpose Float

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 4208

R2 : 4212

R3 : 4216

R4 : 4220

R5 : 4224

R6 : 100

R7 : 200

R8 : 300

R9 : 0

R10 (s1) : 20000

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4160

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 : System

0x800000df

q5.S

.DATA

00001070: A: .WORD 100

00001074: B: .WORD 200

00001078: C: .WORD 300

0000107C: D: .WORD 0

00001080: E: .WORD 0

.TEXT

00001000:E59F1054 LDR R1,=A

00001004:E59F2054 LDR R2,=B

00001008:E59F3054 LDR R3,=C

0000100C:E59F4054 LDR R4,=D

00001010:E3A05D42 LDR R5,=E

00001014:E5916000 LDR R6,[R1]

00001018:E5927000 LDR R7,[R2]

0000101C:E5938000 LDR R8,[R3]

00001020:E5949000 LDR R9,[R4]

00001024:E595A000 LDR R10,[R5]

00001028:E1560007 CMP R6,R7

0000102C:0A000004 BEQ LOOP1

00001030:E1570008 CMP R7,R8

00001034:0A000005 BEQ LOOP2

00001038:E00A0796 MUL R10,R6,R7

0000103C:E585A000 STR R10,[R5]

00001040:EF000011 SWI 0X011

00001044:E0878006 LOOP1: ADD R8,R7,R6

00001048:E5838000 STR R8,[R3]

0000104C:EF000011 SWI 0X011

StackView

MemoryView0

Word Size

8Bit 16Bit 32Bit

00001080

00001080 00004E20 81818181 81818181 81818181

00001090 81818181 81818181 81818181 81818181

000010A0 81818181 81818181 81818181 81818181

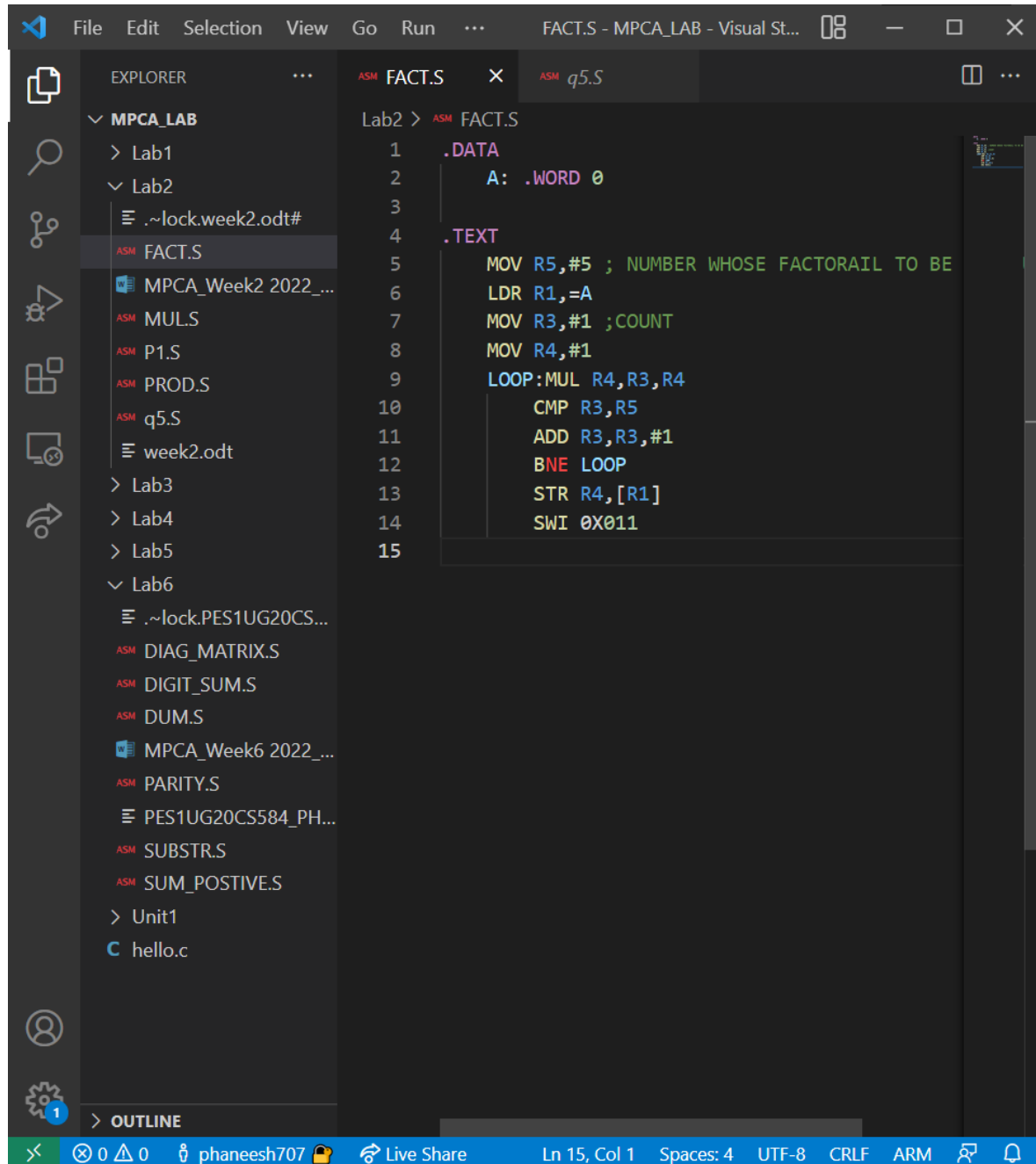
000010B0 81818181 81818181 81818181 81818181

000010C0 81818181 81818181 81818181 81818181

000010D0 81818181 81818181 81818181 81818181

6. Write a program in ARM7TDMI-ISA to find the factorial of a number.

CODE:



The screenshot shows the Visual Studio Code editor with the 'FACT.S' file open. The Explorer sidebar on the left shows a project structure with folders 'Lab1' through 'Lab6' and various files. The main editor area displays the following ARM assembly code:

```
1  .DATA
2      A: .WORD 0
3
4  .TEXT
5      MOV R5,#5 ; NUMBER WHOSE FACTORAIL TO BE
6      LDR R1,=A
7      MOV R3,#1 ;COUNT
8      MOV R4,#1
9      LOOP:MUL R4,R3,R4
10     CMP R3,R5
11     ADD R3,R3,#1
12     BNE LOOP
13     STR R4,[R1]
14     SWI 0X011
15
```

The status bar at the bottom indicates the current position is Line 15, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and the ARM architecture selected.

OUTPUT:

The screenshot displays the ARMSim - The ARM Simulator interface. The window title is "ARMSim - The ARM Simulator Dept. of Computer Science". The menu bar includes File, View, Cache, Debug, Watch, and Help. The main window is divided into several panes:

- RegistersView**: Shows the state of ARM registers. R0-R15 are listed with their values. R15 (pc) is 4132. The CPSR Register shows Negative (N): 0, Zero (Z): 1, Carry (C): 1, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, and CPU Mode: Sy.
- FACT.S**: A tab showing assembly code. The code includes a .DATA section with a word at address 0000102C, and a .TEXT section with instructions: MOV R5, #5; LDR R1, =A; MOV R3, #1; MOV R4, #1; LOOP: MUL R4, R3, R4; CMP R3, R5; ADD R3, R3, #1; BNE LOOP; STR R4, [R1]; and SWI 0X011 at address 00001024.
- MemoryView0**: Shows a memory dump starting at address 0000102C. The word size is set to 32Bit. The memory contains the value 00000078 at address 0000102C, followed by several instances of 81818181.
- OutputView**: An empty pane for displaying program output.