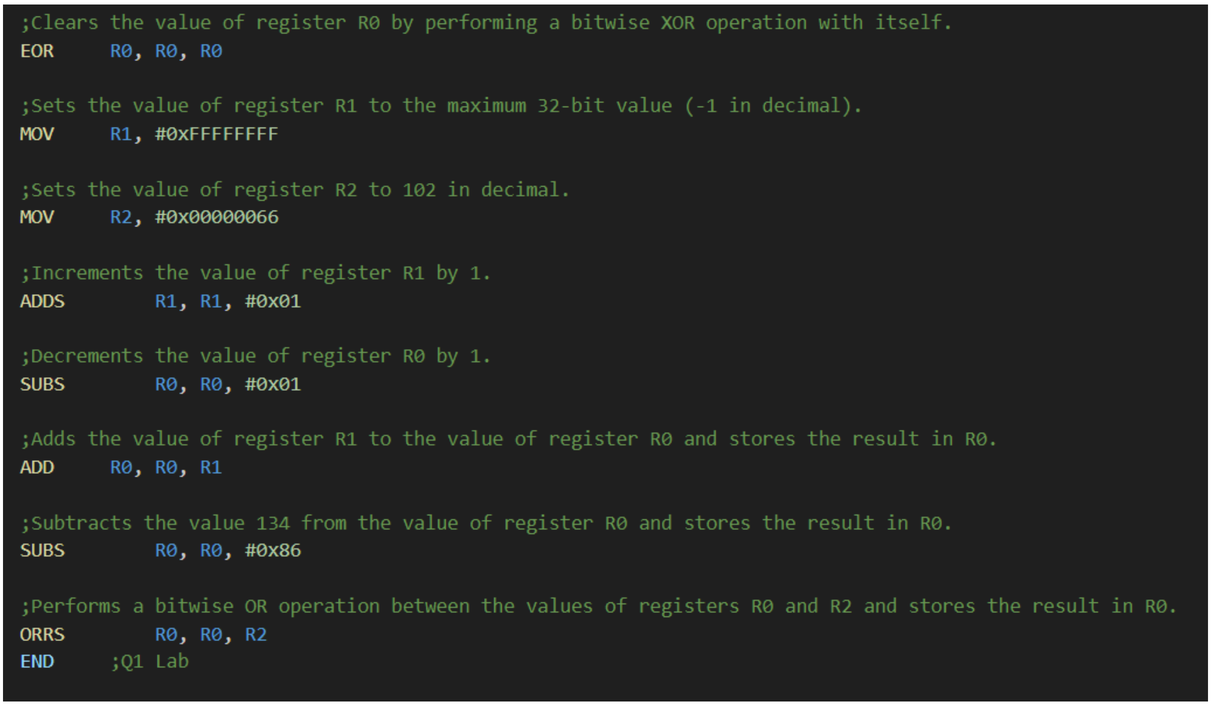
Lab B1

TT0L - GROUP 0

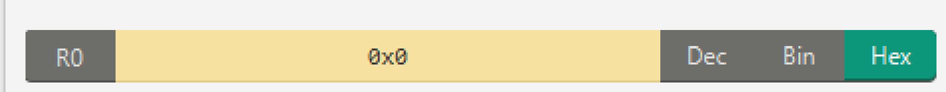
Person 1	1111111111@student.mmu.edu.my
Person 2	1111111111@student.mmu.edu.my

Q1.



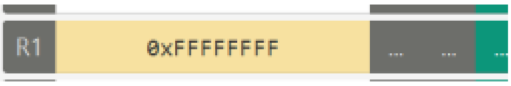
```
;Clears the value of register R0 by performing a bitwise XOR operation with itself.
EOR     R0, R0, R0

;Sets the value of register R1 to the maximum 32-bit value (-1 in decimal).
MOV     R1, #0xFFFFFFFF

;Sets the value of register R2 to 102 in decimal.
MOV     R2, #0x00000066

;Increments the value of register R1 by 1.
ADDS        R1, R1, #0x01

;Decrements the value of register R0 by 1.
SUBS        R0, R0, #0x01

;Adds the value of register R1 to the value of register R0 and stores the result in R0.
ADD     R0, R0, R1

;Subtracts the value 134 from the value of register R0 and stores the result in R0.
SUBS        R0, R0, #0x86

;Performs a bitwise OR operation between the values of registers R0 and R2 and stores the result in R0.
ORRS        R0, R0, R2
END     ;Q1 Lab
```

1. **EOR R0, R0, R0**
   - ○ Clears the value of register R0 by performing a bitwise XOR operation with itself.



R0 0x0

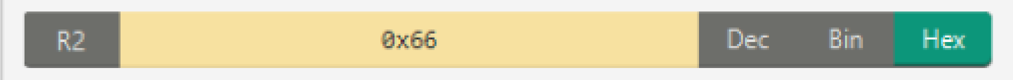
2. **MOV R1, #0xFFFFFFFF**
   - ○ Sets the value of register R1 to the maximum 32-bit value (-1 in decimal).



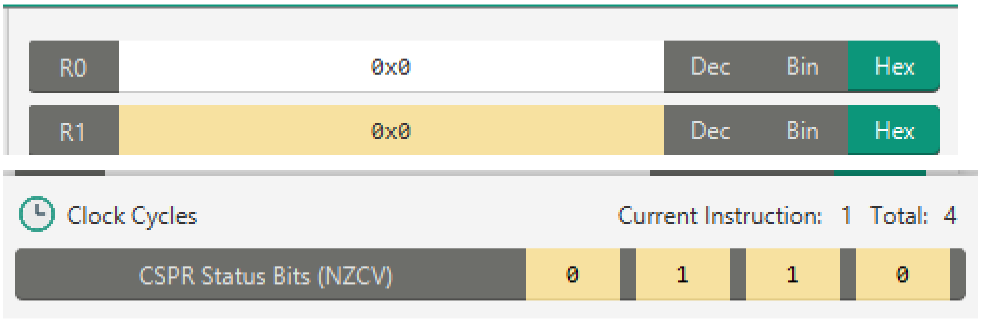
R1 0xFFFFFFFF

3. **MOV R2, #0x00000066**

- Sets the value of register R2 to 102 in decimal.


R2 0x66 Dec Bin Hex

4. ADDS R1, R1, #0x01
    - Increments the value of register R1 by 1.

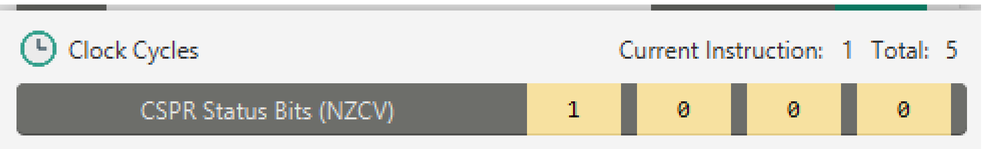

R0 0x0 Dec Bin Hex

R1 0x0 Dec Bin Hex

Clock Cycles Current Instruction: 1 Total: 4

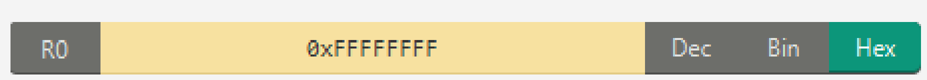
CSPR Status Bits (NZCV) 0 1 1 0

5. SUBS R0, R0, #0x01
    - Decrements the value of register R0 by 1.

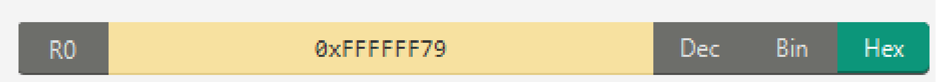

Clock Cycles Current Instruction: 1 Total: 5

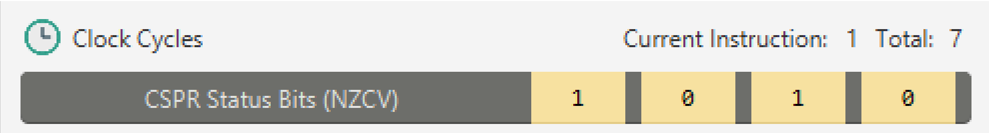
CSPR Status Bits (NZCV) 1 0 0 0

6. ADD R0, R0, R1
    - Adds the value of register R1 to the value of register R0 and stores the result in R0.


R0 0xFFFFFFFF Dec Bin Hex

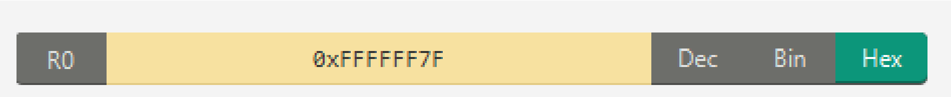
7. SUBS R0, R0, #0x86
    - Subtracts the value 134 from the value of register R0 and stores the result in R0.

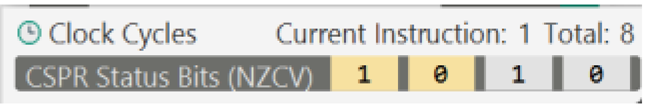

R0 0xFFFFFF79 Dec Bin Hex


Clock Cycles Current Instruction: 1 Total: 7

CSPR Status Bits (NZCV) 1 0 1 0

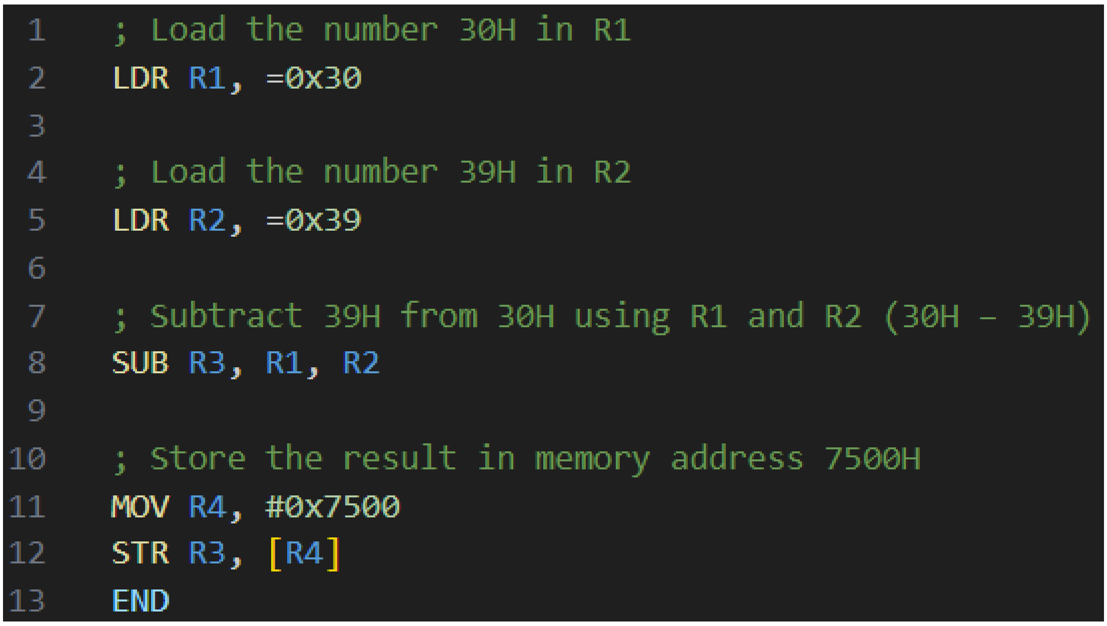
8. ORRS R0, R0, R2
    - Performs a bitwise OR operation between the values of registers R0 and R2 and stores the result in R0.


R0 0xFFFFFF7F Dec Bin Hex

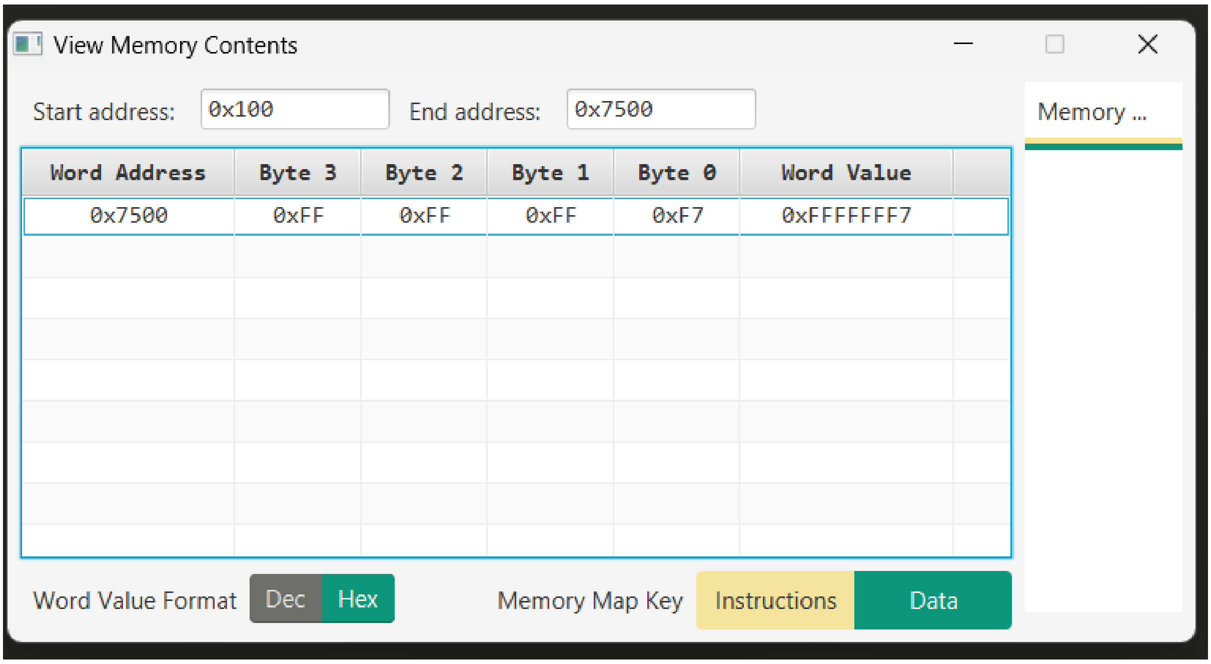

Clock Cycles Current Instruction: 1 Total: 8

CSPR Status Bits (NZCV) 1 0 1 0

Q2.



```
; Load the number 30H in R1
LDR R1, =0x30

; Load the number 39H in R2
LDR R2, =0x39

; Subtract 39H from 30H using R1 and R2 (30H - 39H)
SUB R3, R1, R2

; Store the result in memory address 7500H
MOV R4, #0x7500
STR R3, [R4]
END
```



View Memory Contents

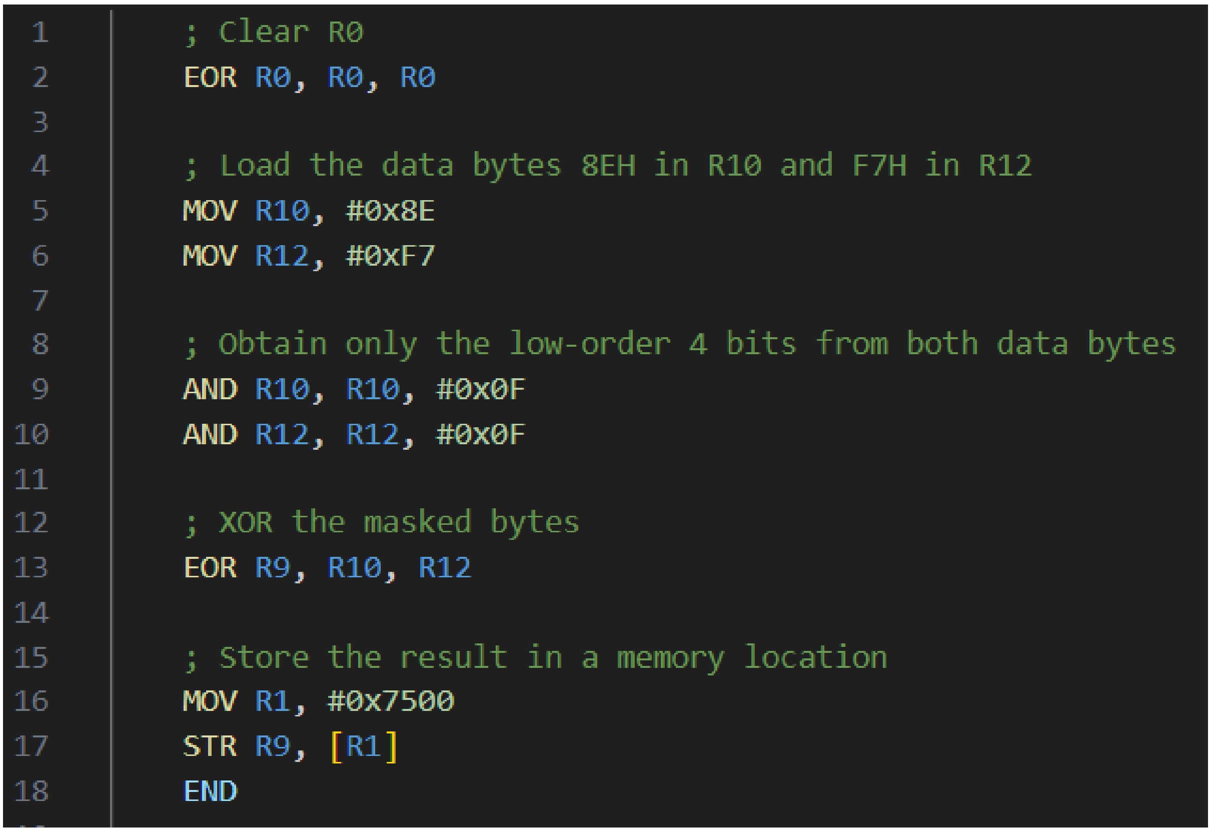
Start address: 0x100

End address: 0x7500

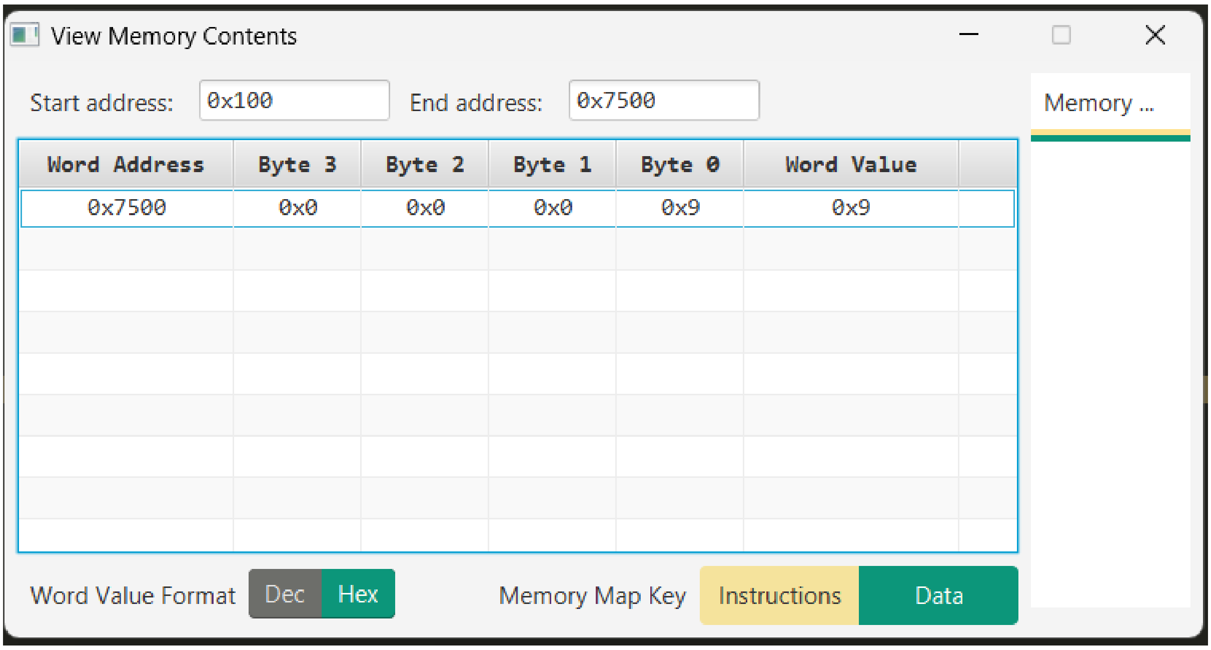
Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x7500	0xFF	0xFF	0xFF	0xF7	0xFFFFFFF7

Word Value Format: Dec [✓ Hex]

Q3.



```
    ; Clear R0
    EOR R0, R0, R0

    ; Load the data bytes 8EH in R10 and F7H in R12
    MOV R10, #0x8E
    MOV R12, #0xF7

    ; Obtain only the low-order 4 bits from both data bytes
    AND R10, R10, #0x0F
    AND R12, R12, #0x0F

    ; XOR the masked bytes
    EOR R9, R10, R12

    ; Store the result in a memory location
    MOV R1, #0x7500
    STR R9, [R1]
    END
```



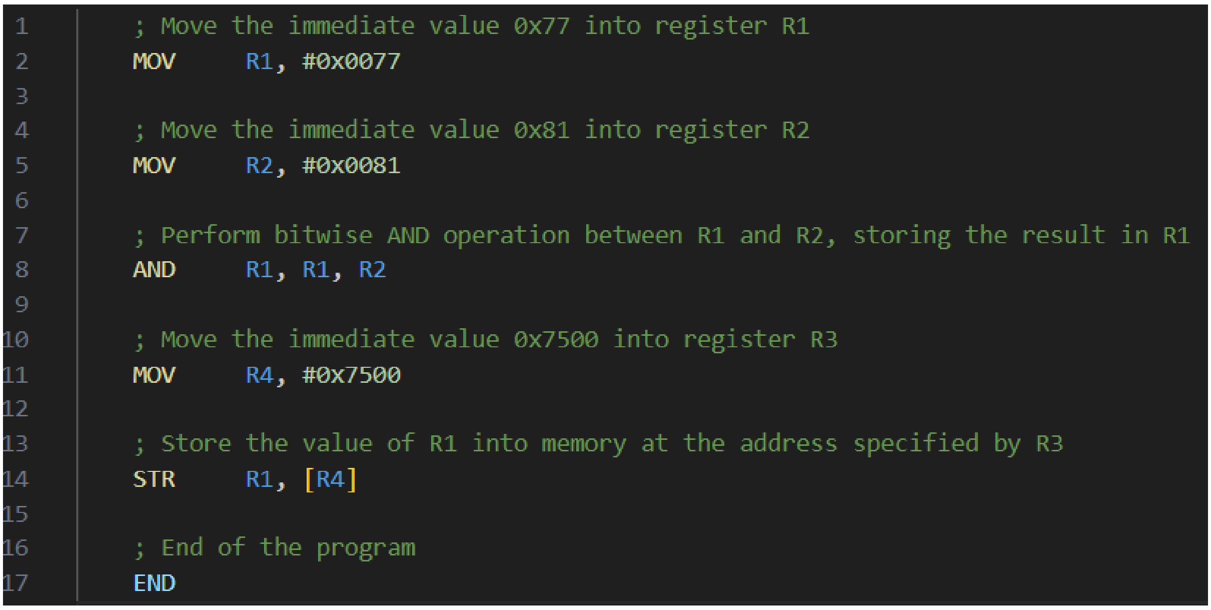
View Memory Contents

Start address: 0x100 End address: 0x7500

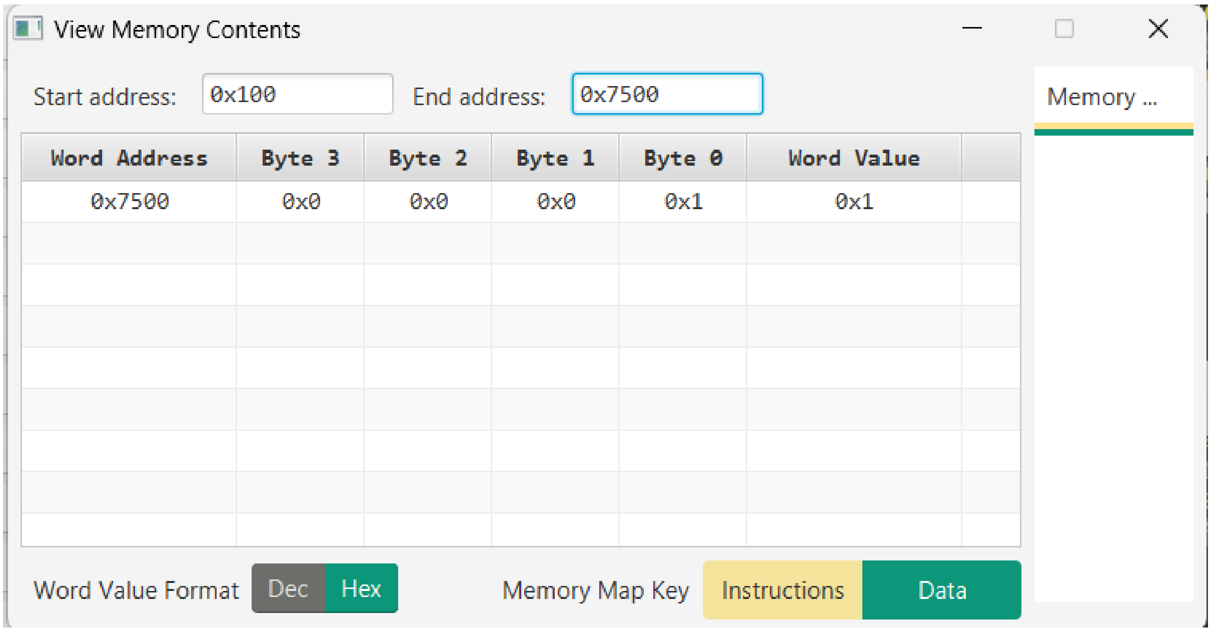
Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x7500	0x0	0x0	0x0	0x9	0x9

Word Value Format: Hex

Q4.



```
; Move the immediate value 0x77 into register R1
MOV     R1, #0x0077

; Move the immediate value 0x81 into register R2
MOV     R2, #0x0081

; Perform bitwise AND operation between R1 and R2, storing the result in R1
AND     R1, R1, R2

; Move the immediate value 0x7500 into register R3
MOV     R4, #0x7500

; Store the value of R1 into memory at the address specified by R3
STR     R1, [R4]

; End of the program
END
```



Start address: 0x100 End address: 0x7500

Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x7500	0x0	0x0	0x0	0x1	0x1

1. MOV R1, #0x0077

- Assign the value 0077H into R1

2. MOV R2, #0x0081

- Assign the value 0081H into R2

3. AND R1, R1, R2

- Add R1 and R2 and store the value into R1

  [R1]=**0100 1101**

[R2]=**1000 0001**

[R1]=**1111 1000**

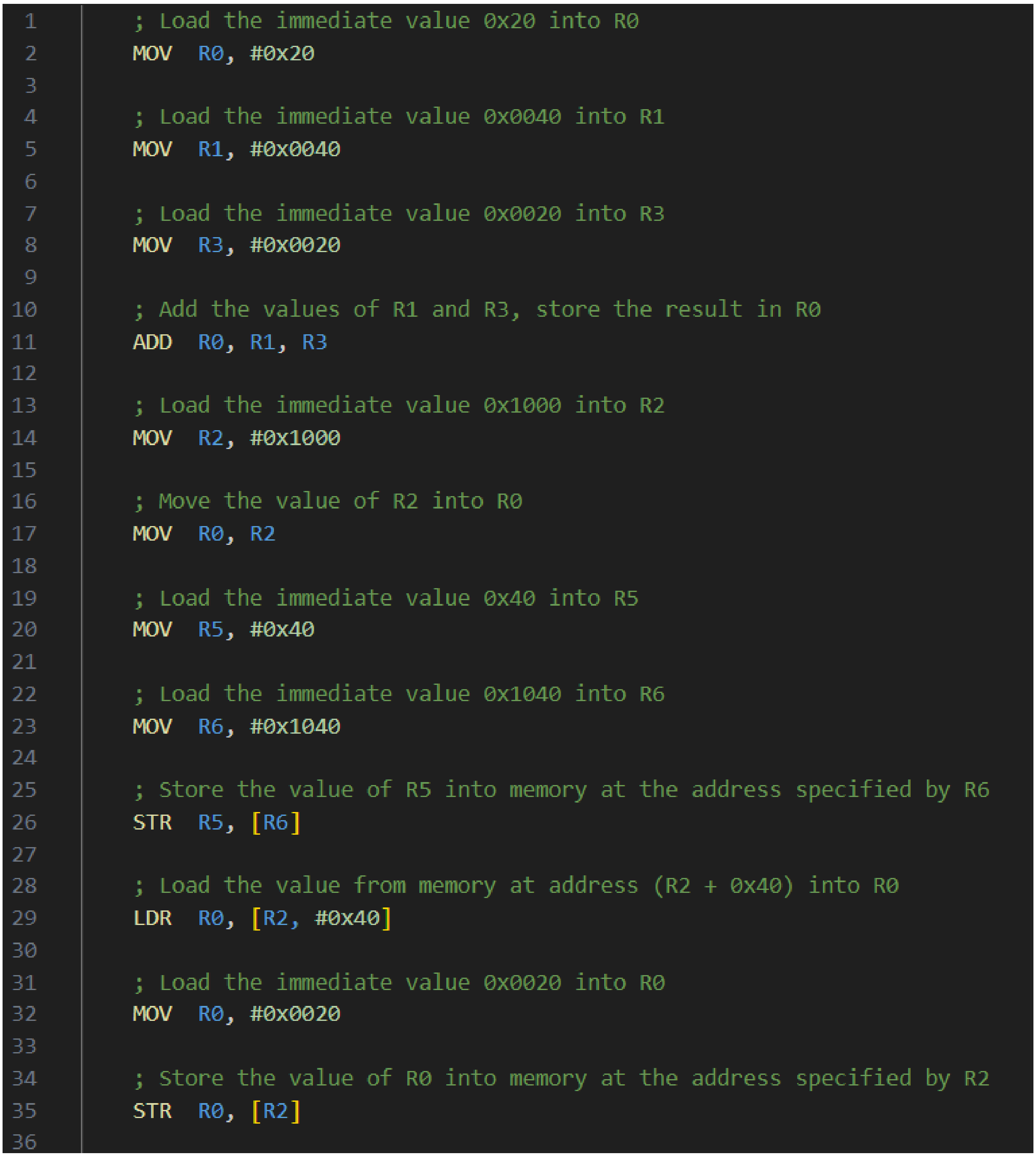
4. MOV R4, #0x7500

- Assign the value 7500H into R4

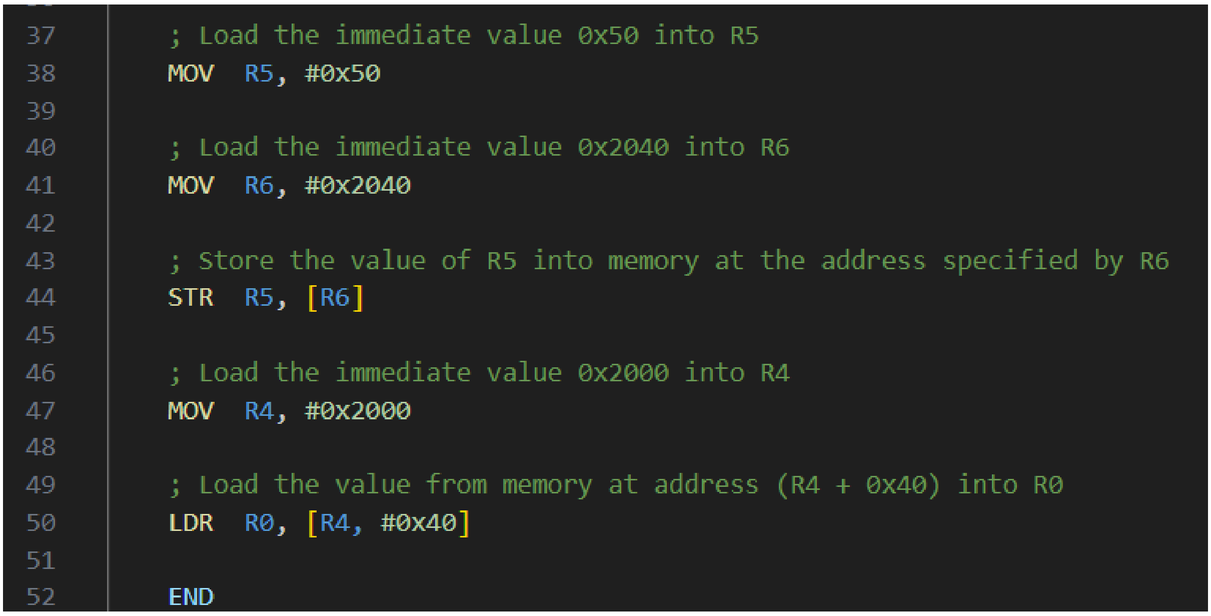
5. STR R1, [R4]

- Store the value of R1 into memory address specified by R4

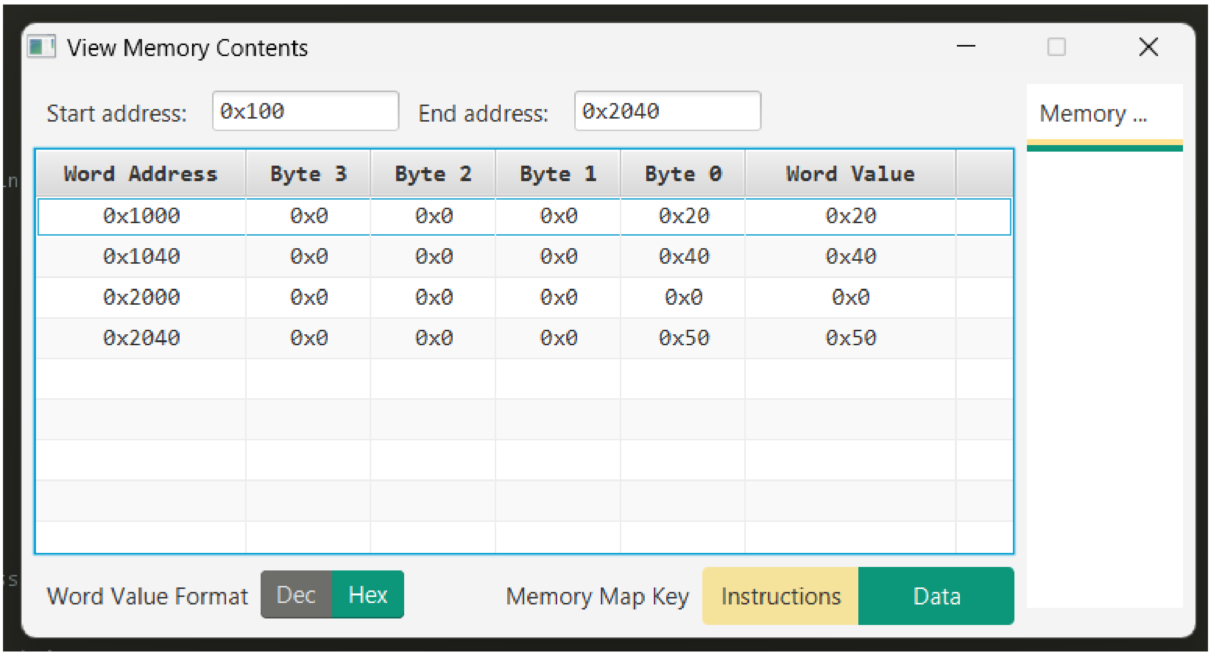
Q5. Given the following register and memory values, what values do the following ARM instructions load into R0?



```
; Load the immediate value 0x20 into R0
MOV  R0, #0x20

; Load the immediate value 0x0040 into R1
MOV  R1, #0x0040

; Load the immediate value 0x0020 into R3
MOV  R3, #0x0020

; Add the values of R1 and R3, store the result in R0
ADD  R0, R1, R3

; Load the immediate value 0x1000 into R2
MOV  R2, #0x1000

; Move the value of R2 into R0
MOV  R0, R2

; Load the immediate value 0x40 into R5
MOV  R5, #0x40

; Load the immediate value 0x1040 into R6
MOV  R6, #0x1040

; Store the value of R5 into memory at the address specified by R6
STR  R5, [R6]

; Load the value from memory at address (R2 + 0x40) into R0
LDR  R0, [R2, #0x40]

; Load the immediate value 0x0020 into R0
MOV  R0, #0x0020

; Store the value of R0 into memory at the address specified by R2
STR  R0, [R2]
```



```
        ; Load the immediate value 0x50 into R5
        MOV   R5, #0x50

        ; Load the immediate value 0x2040 into R6
        MOV   R6, #0x2040

        ; Store the value of R5 into memory at the address specified by R6
        STR   R5, [R6]

        ; Load the immediate value 0x2000 into R4
        MOV   R4, #0x2000

        ; Load the value from memory at address (R4 + 0x40) into R0
        LDR   R0, [R4, #0x40]

        END
```



Start address: 0x100 End address: 0x2040

Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x1000	0x0	0x0	0x0	0x20	0x20
0x1040	0x0	0x0	0x0	0x40	0x40
0x2000	0x0	0x0	0x0	0x0	0x0
0x2040	0x0	0x0	0x0	0x50	0x50

i)MOV R0, #0x20

R0 = 0x20

ii)ADD R0, R1, R3

R0 = 0x60

iii)MOV R0, R2

R0 = 0x1000

iv)LDR R0, [R2, #0x40]

R0 = 0x40

v)STR R0, [R2]

R0 = 0x20

vi)LDR R0, [R4, R1]

R0 = 0x50

Q6. Specify the flag status if the following ARM instructions are executed:

EORS R0, R0, R0:

NZCV = 0100

SUB R0, R0, #0x01:

NZCV = 0100

MOV R1, #0xFF:

NZCV = 0100

ADDS R1, R1, #0x01:

NZCV = 0000