

registers

Register	Value
R0	0x00000019
R1	0x20000619
R2	0x20000634
R3	0x00000200
R4	0x20000063
R5	0x20000008
R6	0x00000000
R7	0x00000000
R8	0x20000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x20000049
R13 (SP)	0x20000619
R14 (LR)	0x00000200
R15 (PC)	0x00000490
PSR	0x01000000

Disassembly

0x00000490 B108 CBZ r0,0x00000496

0x00000492 2000 MOVs r0,#0x00

0x00000494 BD1C POP {r2-r4,pc}

0x00000496 4620 MOV r0,r4

code.s

startup\_stm32f412xx.s

28

29 ADD R1, R1, #1

30 SUB R2, R2, #1

31 CMP R1, R2

32 BNE LOOP

33

34 PALINDROME

35 MOV R0, #1

36 B END

37

38 NOT\_PALINDROME

39 MOV R0, #0

40 B END

41

42 END

43

44

45 AREA DATA, DATA, READWRITE

46 string DCB "madam", 0

47

48

command

Watch 2

Running with Code Size Limit: 32K

Load "C:\Users\user\Downloads\Assignment-1 B22CS001\Code 4\Objects\code 4.axf"

IS 2, 'r4

IS 2, 'R0

Name	Value	Type
r4	0x00000061	ulong
R0	0x00000001	ulong
<Enter expression>		

Watch 2		
Name	Value	Type
r4	0x00000061	ulong
R0	0x00000001	ulong
<Enter expression>		

# EEP3020 : Digital Systems Lab

## Assignment - I

---

**Name :** Abhay Kashyap

**Roll no :** B22CS001

---

### **Question - 1. Check Greater between two input numbers**

```
                PRESERVE8
                TTL      TEXT
                GLOBAL   main

                AREA     Data, DATA, READWRITE
                ALIGN
NUM1            DCD      7
NUM2            DCD      10

                AREA     Compare, CODE, READONLY
                ENTRY

main
    LDR          r0, =NUM1
    LDR          r1, =NUM2
    LDR          r2, [r0]
    LDR          r3, [r1]

    CMP          r2, r3
    BGT          num1_greater
    BLT          num2_greater
    BEQ          equal

num1_greater
    MOV          r4, r2
    B            end_comparison

num2_greater
    MOV          r4, r3
    B            end_comparison

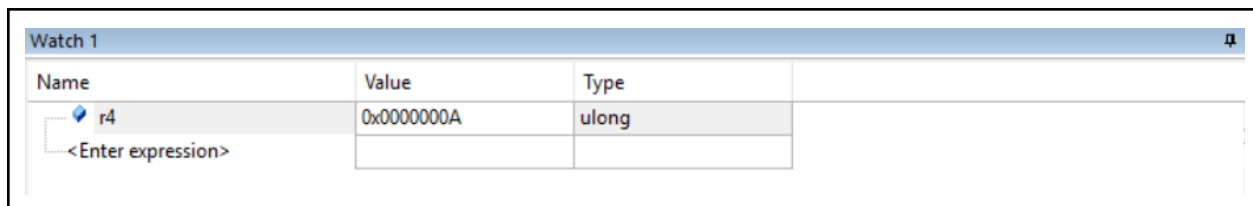
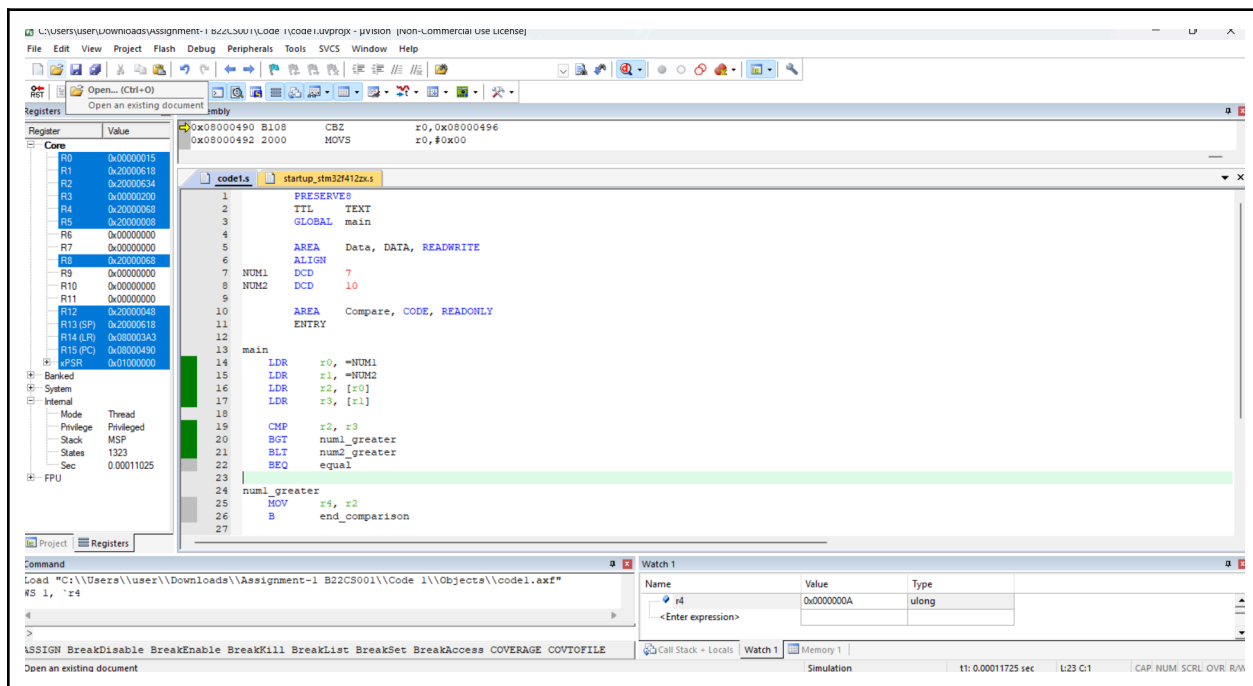
equal
    MOV          r4, r2

end_comparison
    MOV          r7, #1
    SWI          0

END
```

## Explanation :

The ARM assembly code compares two numbers (7 and 10, stored in **NUM1** and **NUM2**) and stores the greater of the two in register **r4**. If both numbers are equal, **r4** is set to the value of **NUM1**. After the comparison, the program makes a system call to exit. The result of the comparison is controlled by conditional branching (**BGT**, **BLT**, **BEQ**).



**Question - 2. Calculate the minimum one between elements of an array**

