Title	**Special Topics in Algorithms**	Number	**CS7xxx**
Department	Computer Science	L-T-P [C]	3–0–0 [3]
Offered for	B. Tech., M.Tech., PhD	Type	Elective
Prerequisite	Algorithm Design and Analysis, Maths for Computing	Antirequisite / Preferred Knowledge	None

Objectives

1. The objective of the course is to introduce several advanced algorithmic techniques.

Learning Outcomes

Students will gain the ability to:

1. Learn a new set of techniques to cope with NP-hard problems.
2. Identify novel and significant open research questions in the field.

Contents

Parameterized Algorithms [13 lectures]: Introduction to Parameterized Complexity and basics [2 Lectures]; Branching [4 Lectures]; Iterative Compression [3 Lectures]; Kernelization [4 Lectures]
Approximation Algorithms: [10 lectures]: Greedy Algorithm – Load Balancing, Center Selection Problem, Set Cover [5 Lectures]; The Pricing Method: Vertex Cover, Linear Programming and Rounding: An application to Vertex Cover, Knapsack [5 Lectures]
Randomized Algorithms [10 lectures]: Contention Resolution, Global Mincut, Random Variables and Expectations, Max-3-SAT approximation [7 Lectures]; Color Coding [3 Lectures]
Exact Exponential Time Algorithms [7 lectures]: Exact Algorithms for Coloring, SAT, Directed Feedback Arc Set, Max-Cut, Monotone-Local-Search, Or some other topics of contemporary interest.
Streaming Algorithms [2 lectures]: Introduction to streaming algorithms and its application to some graph theoretic problems.

Textbooks

1. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh (2015): Parameterized Algorithms, Springer.
2. Jon Kleinberg, Eva Tardos (2005), Algorithm Design, Pearson Education, 1st Edition.
3. Fedor V. Fomin, Dieter Kratsch (2010), Exact Exponential Time Algorithms, An EATCS Series, Springer.

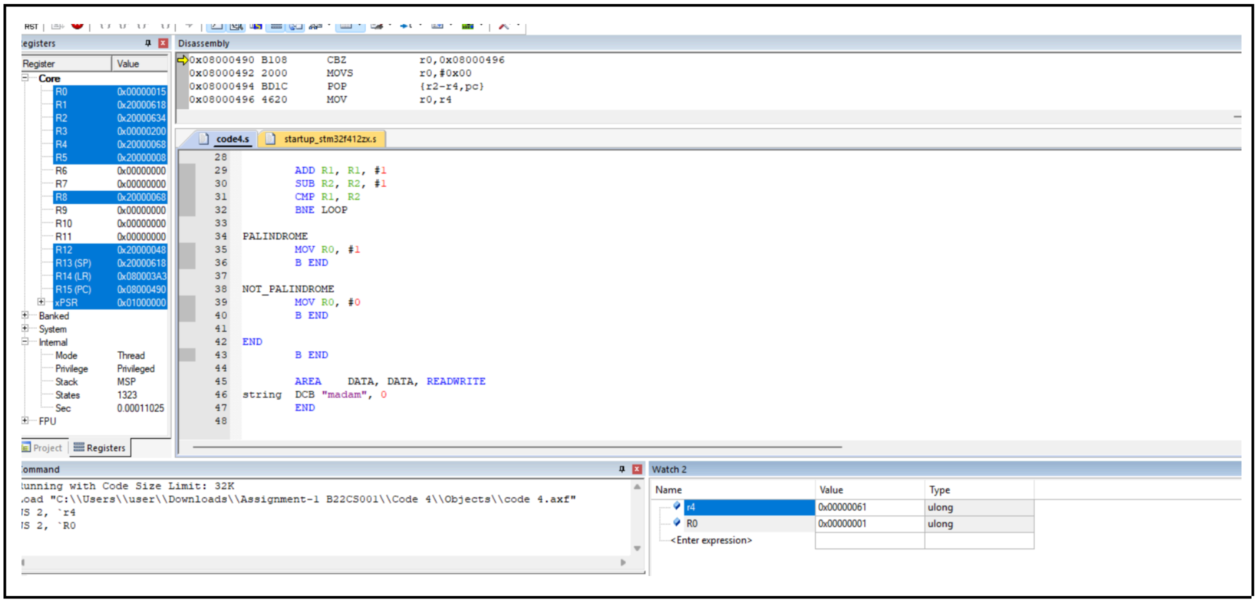
Self Learning Material

1. https://www.youtube.com/watch?v=Ex8TueBsF1g&list=PLhkiT_RYTEU0gpi97fqjtaHy9Gk47oF85&index=1
2. https://sites.google.com/view/sakethome/teaching/parameterized-complexity?authuser=0
3. https://www.youtube.com/watch?v=S8Acu3EpvsE&list=PLhkiT_RYTEU2itsMqCNdXUg4cdFUWJn3-&index=4
4. https://www.youtube.com/watch?v=jNfQ3GZlrjM&list=PLhkiT_RYTEU3vSaVleEm_-bIPBzCqRQHK

Courses Offered by

Department of Computer Science and Engineering

Course - List



egisters

Register	Value
Core	
R0	0x00000015
R1	0x20000618
R2	0x20000634
R3	0x00000200
R4	0x20000068
R5	0x20000008
R6	0x00000000
R7	0x00000000
R8	0x20000068
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x20000048
R13 (SP)	0x20000618
R14 (LR)	0x080003A3
R15 (PC)	0x08000490
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	1323
Sec	0.00011025
FPU	

Disassembly

```
0x08000490 B108      CBZ          r0,0x08000496
0x08000492 2000      MOVS         r0,#0x00
0x08000494 BD1C      POP          {r2-r4,pc}
0x08000496 4620      MOV          r0,r4
```

code4.s   startup_stm32f412zx.s

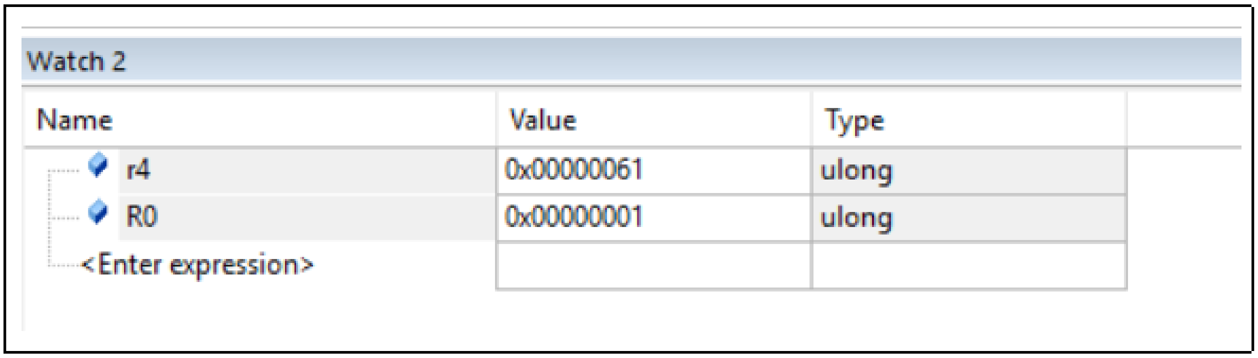
```

        ADD R1, R1, #1
        SUB R2, R2, #1
        CMP R1, R2
        BNE LOOP

PALINDROME
        MOV R0, #1
        B END

NOT_PALINDROME
        MOV R0, #0
        B END

END
        B END

        AREA    DATA, DATA, READWRITE
string  DCB "madam", 0
        END

```

ommand

```
unning with Code Size Limit: 32K
oad "C:\\Users\\user\\Downloads\\Assignment-1 B22CS001\\Code 4\\Objects\\code 4.axf"
S 2, `r4
S 2, `R0
```

Watch 2

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*

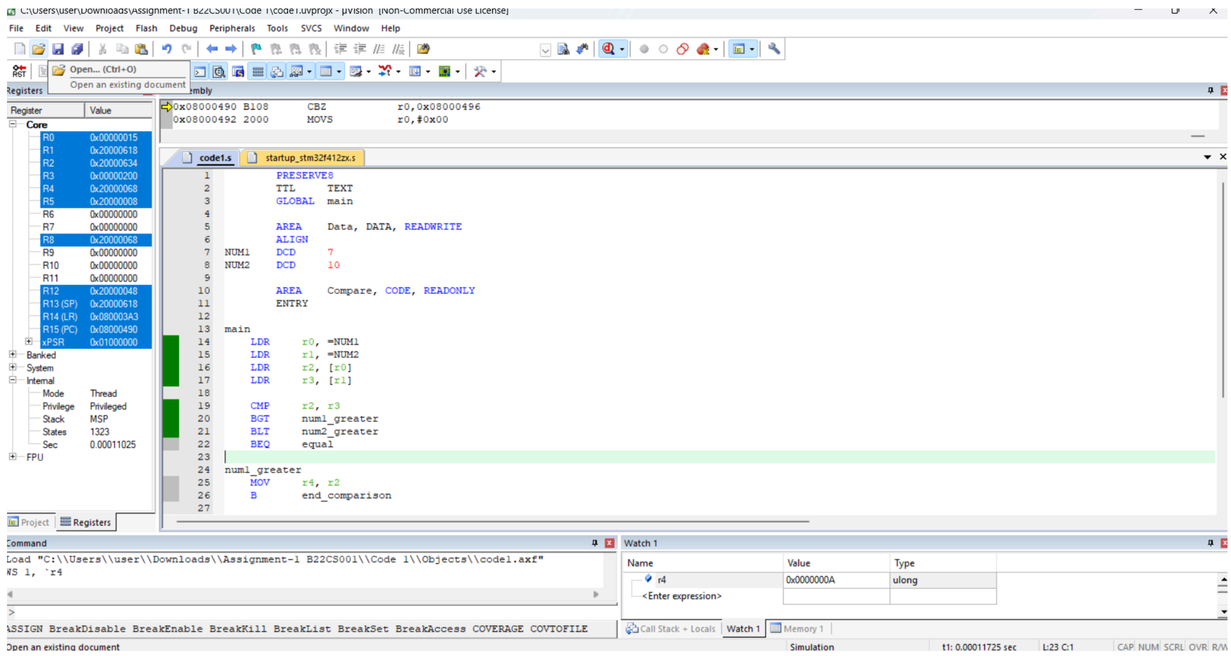
Roll no : *B22CS001*

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).


C:\Users\user\Downloads\Assignment-1 B22CS001\Code 1\code1.uvprojx - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Open... (Ctrl+O)
Open an existing document
Registers
Disassembly
code1.s
startup_stm32f412zx.s
Project
Registers
Command
Call Stack + Locals
Watch 1
Memory 1

```
0x08000490 B108      CBZ        r0,0x08000496
0x08000492 2000      MOVS       r0,#0x00
```

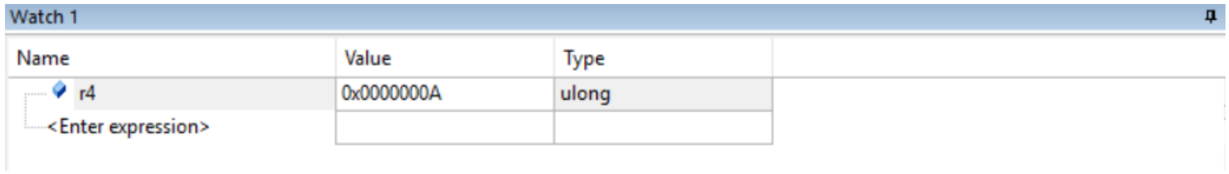
Register	Value
Core	
R0	0x00000015
R1	0x20000618
R2	0x20000634
R3	0x00000200
R4	0x20000068
R5	0x20000008
R6	0x00000000
R7	0x00000000
R8	0x20000068
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x20000048
R13 (SP)	0x20000618
R14 (LR)	0x080003A3
R15 (PC)	0x08000490
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	1323
Sec	0.00011025
FPU	

```
        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

```

```
Load "C:\\Users\\user\\Downloads\\Assignment-1 B22CS001\\Code 1\\Objects\\code1.axf"
WS 1, `r4
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Value	Type
r4	0x0000000A	ulong
<Enter expression>		

Open an existing document Simulation t1: 0.00011725 sec L:23 C:1 CAP NUM SCRL OVR R/W



Name	Value	Type
r4	0x0000000A	ulong
<Enter expression>		

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

```
      PRESERVE8
      TTL     TEXT
      GLOBAL  main

      AREA    Data, DATA, READWRITE
      ALIGN
ARRAY DCD     5, 10, 3, 7, 2
LEN   DCD     5
```