



Department of Computer Science & Engineering
Microprocessor & Computer Architecture Lab

Lab 2 Submission Format

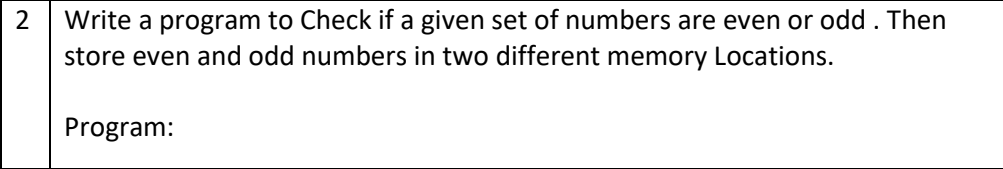
UE22CS251B

- 1 Write a program to add two numbers by reading them from memory and store the result back to the memory.

Program:

```
week3_q1.s
1  .DATA
2  A: .WORD 10,20
3  RES: .WORD 0
4  .TEXT
5  LDR R0 ,=A
6  LDR R2 ,=RES
7  LDR R1,[R0,#4]
8  LDR R3,[R0,#4]
9  ADD R1,R1,R3
10 STR R1,[R2]
11 SWI 0x100
12 .END
13
14
```

Output Screen Shot:



```

week3_q2.s
1  .DATA
2  A: .WORD 10,3,41,55,30,22,11,5,83,90
3  ODD: .WORD 0
4  EVEN: .WORD 0
5
6  .TEXT
7  LDR R0,=A
8  LDR R1,=ODD
9  LDR R2,=EVEN
10 LDR R3,=10
11
12 CHECK_NO:
13 LDR R4, [R0]
14 AND R4,R4,#1
15
16 CMP R4,#1
17 BEQ ODD_NO
18
19 STR R4,[R2],#4
20 B END_PROGRAM
21
22 ODD_NO:
23 STR R4,[R1],#4
24
25 END_PROGRAM:
26 ADD R0,R0,#4
27 SUBS R3,R3,#1
28 BNE CHECK_NO
29
30

```

Output Screen Shot:

ARMv8 - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

Registers

General Purpose Floating Point

Headecimal

Unsigned Decimal

Signed Decimal

R0 :00001060

R1 :00000000

R2 :00001060

R3 :00000000

R4 :00000000

R5 :00000000

R6 :00000000

R7 :00000000

R8 :00000000

R9 :00000000

R10(L):00000000

R11(fp):00000000

R12(lp):00000000

R13(sp):00005400

R14(lr):00000000

R15(pc):00001034

CPSR Register

Negative(N):0

Zero(Z):1

Carry(C):1

Overflow(O):0

IRQ Disable:1

FIQ Disable:1

Thumb(T):0

CPU Mode :system

0x600000df

week3_q4.s

```

.DATA
00001038: A: .WORD 10, 50, 41, 55, 30, 20, 11, 5, 100, 77
00001040: LARGEST: .WORD 0

.TEXT
00001000: E59F0028 LDR R0, -A
00001004: E59B1000 LDR R1, [R0]
00001008: E3A0100A LDR R3, -10

0000100C: LARGEST: MO:
0000100C: E59B2000 LDR R2, [R0]
00001010: E1510002 CMP R1, R2
00001014: 1A600000 B.T CHECK_NEXT
00001018: E1A01002 MOV R1, R2

0000101C: CHECK_NEXT:
0000101C: E2800004 ADD R0, R0, #4
00001020: E2513001 STRS R3, R3, #1
00001024: 1AFFFFF9 BNE LARGEST_MO
00001028: E59F2004 LDR R4, -LARGEST
0000102C: E59B1000 STR R1, [R2]

```

Memory View2

Word Size

8Bit 16Bit 32Bit

00000000

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00000030 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00000060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00000090 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

27°C Sunny

Search

1437

25-01-2024

3

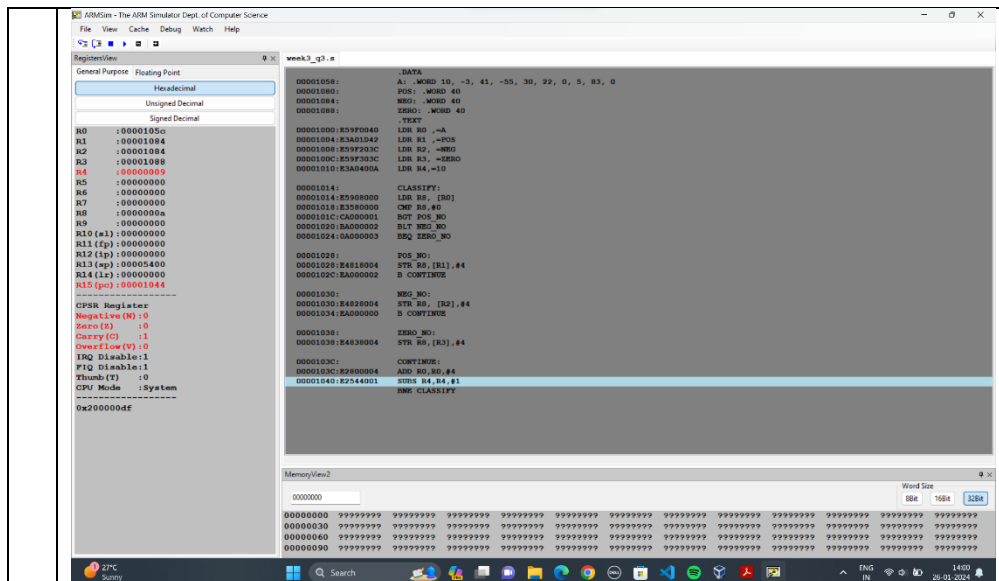
Write a program to Classify the given set of numbers as positive, negative or zero and also store them in different memory locations.

Program:

week3_q3.s

```
1  .DATA
2  A: .WORD 10, -3, 41, -55, 30, 22, 0, 5, 83, 0
3  POS: .WORD 40
4  NEG: .WORD 40
5  ZERO: .WORD 40
6  .TEXT
7  LDR R0, =A
8  LDR R1, =POS
9  LDR R2, =NEG
10 LDR R3, =ZERO
11 LDR R4, =10
12
13 CLASSIFY:
14 LDR R8, [R0]
15 CMP R8, #0
16 BGT POS_NO
17 BLT NEG_NO
18 BEQ ZERO_NO
19
20 POS_NO:
21 STR R8, [R1], #4
22 B CONTINUE
23
24 NEG_NO:
25 STR R8, [R2], #4
26 B CONTINUE
27 |
28 ZERO_NO:
29 STR R8, [R3], #4
30
31 CONTINUE:
32 ADD R0, R0, #4
33 SUBS R4, R4, #1
34 BNE CLASSIFY
35
```

Output Screen Shot:



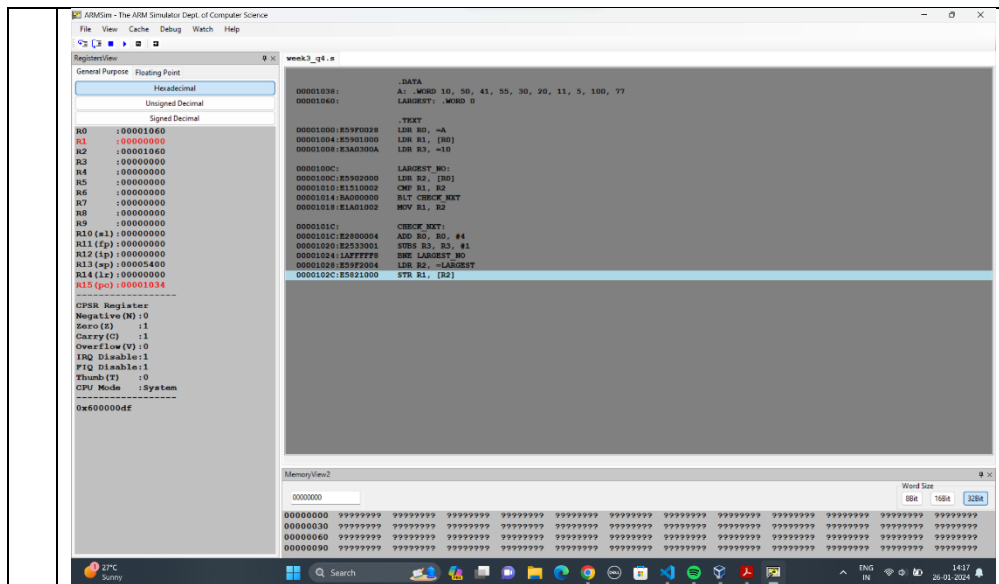
- 4 Write a program to find the largest number from a given set of numbers.
Program:

```

1  week3_q4.s
2  .DATA
3  A: .WORD 10, 50, 41, 55, 30, 20, 11, 5, 100, 77
4  LARGEST: .WORD 0
5
6  .TEXT
7  LDR R0, =A
8  LDR R1, [R0]
9  LDR R3, =10
10
11  LARGEST_NO:
12  LDR R2, [R0]
13  CMP R1, R2
14  BLT CHECK_NXT
15  MOV R1, R2
16
17  CHECK_NXT:
18  ADD R0, R0, #4
19  SUBS R3, R3, #1
20  BNE LARGEST_NO
21  LDR R2, =LARGEST
22  STR R1, [R2]

```

Output Screen Shot:



5 Assignment Questions:

i)

Write a program to add array of ten 8-bit numbers taking data from memory location stored as byte data (use .byte to store the data instead of .word)

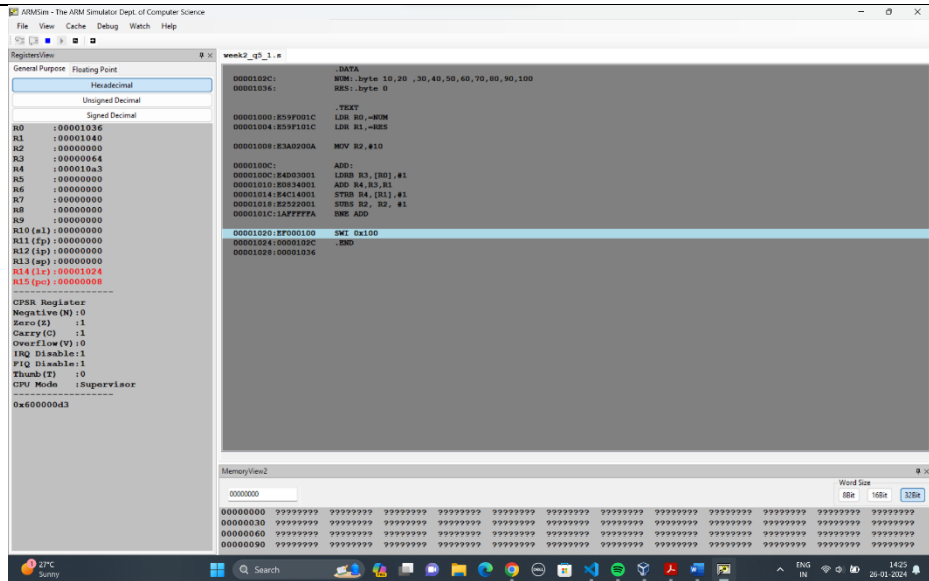
Program:

```

1  .DATA
2  NUM:.byte 10,20 ,30,40,50,60,70,80,90,100
3  RES:.byte 0
4
5  .TEXT
6  LDR R0,=NUM
7  LDR R1,=RES
8
9  MOV R2,#10
10
11  ADD:
12  LDRB R3,[R0],#1
13  ADD R4,R3,R1
14  STRB R4,[R1],#1
15  SUBS R2, R2, #1
16  BNE ADD
17
18  SWI 0x100
19  .END

```

Output Screen Shot:



ii)

Generate Fibonacci Series and store them in an array / memory location.

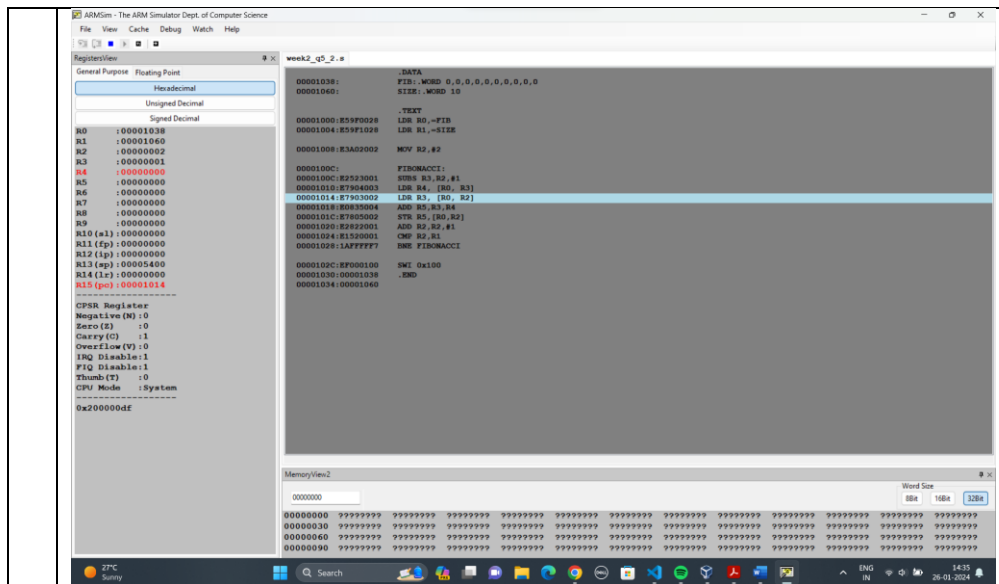
Program:

```

week2_q5_2.s
1  .DATA
2  FIB:.WORD 0,0,0,0,0,0,0,0,0,0
3  SIZE:.WORD 10
4
5  .TEXT
6  LDR R0,=FIB
7  LDR R1,=SIZE
8
9  MOV R2,#2
10
11 FIBONACCI:
12 SUBS R3,R2,#1
13 LDR R4, [R0, R3]
14 LDR R3, [R0, R2]
15 ADD R5,R3,R4
16 STR R5,[R0,R2]
17 ADD R2,R2,#1
18 CMP R2,R1
19 BNE FIBONACCI
20
21 SWI 0x100
22 .END

```

Output Screen Shot:



Note:

- Link to upload the file:
 - Will be provided by the respective Theory Teacher
- Upload PDF only.
- Save your file with your SRN _ Name