# Microprocessor and Computer Architecture Laboratory

# UE19CS256

## 4th Semester, Academic Year 2020-21

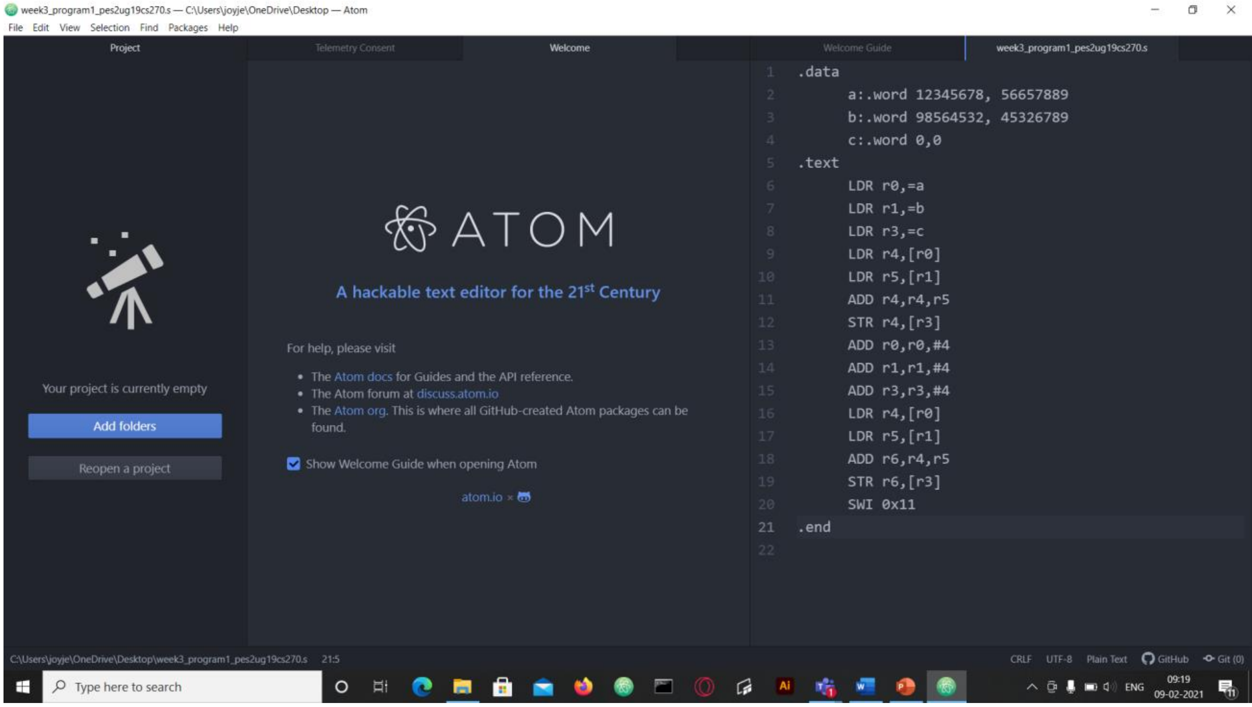
Date:

| | | |
|---|---|---|
| Name: OP Joy Jefferson | SRN:PES2UG19CS270 | Section:E |

Week#____3_______ Program Number: ____1__

## Write an ALP to add two 64 bit numbers loaded from memory and store the result in memory.

I. ARM Assembly Code for the program.



week3_program1_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Telemetry Consent
Welcome
Welcome Guide
week3_program1_pes2ug19cs270.s
Your project is currently empty
Add folders
Reopen a project
ATOM
A hackable text editor for the 21st Century
atom.io
C:\Users\joyje\OneDrive\Desktop\week3_program1_pes2ug19cs270.s 21:5
CRLF
UTF-8
Plain Text
GitHub
Git (0)
Type here to search
ENG
09:19
09-02-2021

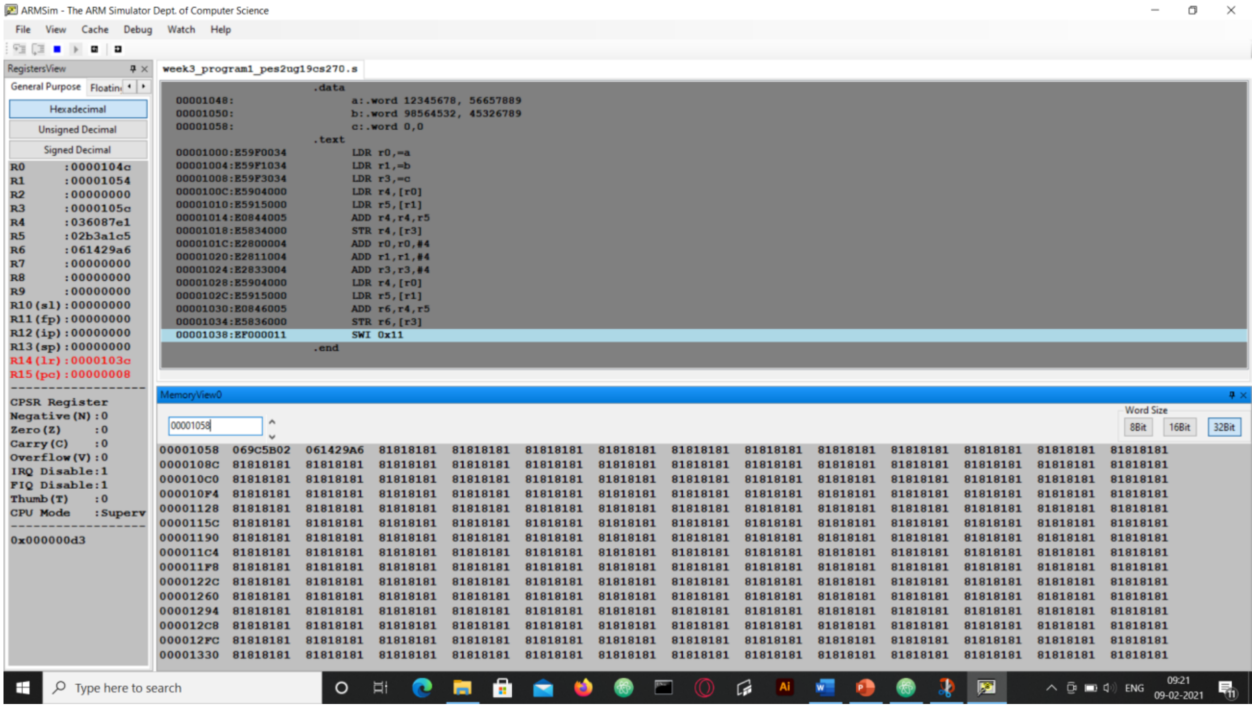
For help, please visit

- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io
- The Atom org. This is where all GitHub-created Atom packages can be found.

☑ Show Welcome Guide when opening Atom

```
.data
      a:.word 12345678, 56657889
      b:.word 98564532, 45326789
      c:.word 0,0
.text
      LDR r0,=a
      LDR r1,=b
      LDR r3,=c
      LDR r4,[r0]
      LDR r5,[r1]
      ADD r4,r4,r5
      STR r4,[r3]
      ADD r0,r0,#4
      ADD r1,r1,#4
      ADD r3,r3,#4
      LDR r4,[r0]
      LDR r5,[r1]
      ADD r6,r4,r5
      STR r6,[r3]
      SWI 0x11
.end
```

## II. Output Screen Shot (One Example of your choice)



```
R0       :0000104c
R1       :00001054
R2       :00000000
R3       :0000105c
R4       :036087e1
R5       :02b3a1c5
R6       :061429a6
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :0000103c
R15(pc)  :00000008
------------------
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   :Superv
------------------
0x000000d3
```

```
                        .data
00001048:                      a:.word 12345678, 56657889
00001050:                      b:.word 98564532, 45326789
00001058:                      c:.word 0,0
                        .text
00001000:E59F0034              LDR r0,=a
00001004:E59F1034              LDR r1,=b
00001008:E59F3034              LDR r3,=c
0000100C:E5904000              LDR r4,[r0]
00001010:E5915000              LDR r5,[r1]
00001014:E0844005              ADD r4,r4,r5
00001018:E5834000              STR r4,[r3]
0000101C:E2800004              ADD r0,r0,#4
00001020:E2811004              ADD r1,r1,#4
00001024:E2833004              ADD r3,r3,#4
00001028:E5904000              LDR r4,[r0]
0000102C:E5915000              LDR r5,[r1]
00001030:E0846005              ADD r6,r4,r5
00001034:E5836000              STR r6,[r3]
00001038:EF000011              SWI 0x11
                        .end
```

```
00001058
00001058 069C5B02 061429A6 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000108C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010C0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010F4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001128 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000115C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001190 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000011C4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000011F8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000122C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001260 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001294 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000012C8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000012FC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001330 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

## III. Output Table for the program

|  | **a: .word 12345678, 56657889**<br>**b: .word 98764532, 45326789** |  |
|---|---|---|
|  | Upper 32 bits | **Lower 32 bits** |
| **a: .word** | 56657889<br>**(036087DF)** | 12345678<br>**(00BC614E)** |
| **b: .word** | 45326789<br>**(02B3A1C5)** | 98764532<br>**(05E306F4)** |
| **c: .word** | 101984678<br>**(061429A6)** | 111110210<br>**(069C5B02)** |
|  |  |  |

**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

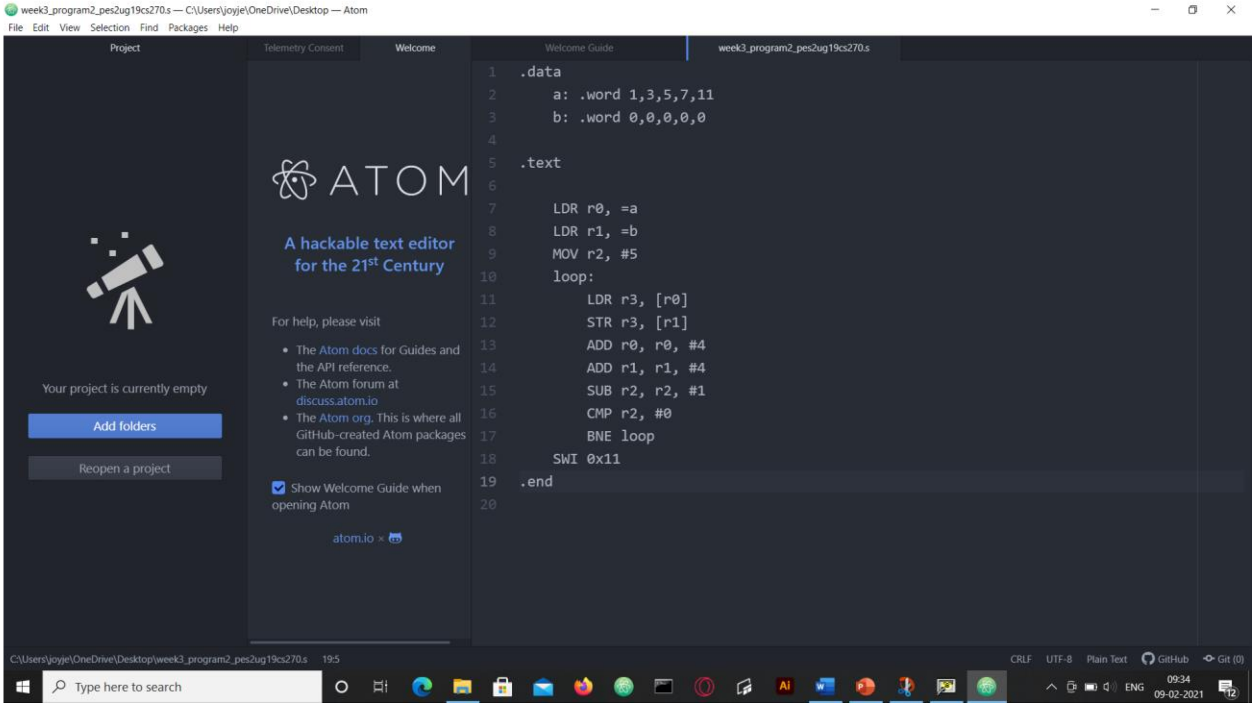
Date:

| | | |
|---|---|---|
| Name: OP JOY JEFFERSON | SRN:PES2UG19CS270 | Section:E |

Week#____3_______ Program Number: ____2__

## Write an ALP to copy n numbers from Memory Location A to Memory Location B

I. ARM Assembly Code for the program.



week3_program2_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Telemetry Consent
Welcome
Welcome Guide
week3_program2_pes2ug19cs270.s
Your project is currently empty
Add folders
Reopen a project
ATOM
A hackable text editor for the 21st Century
atom.io × 
C:\Users\joyje\OneDrive\Desktop\week3_program2_pes2ug19cs270.s 19:5
CRLF UTF-8 Plain Text GitHub Git (0)
Type here to search
ENG
09:34
09-02-2021

For help, please visit

- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io
- The Atom org. This is where all GitHub-created Atom packages can be found.

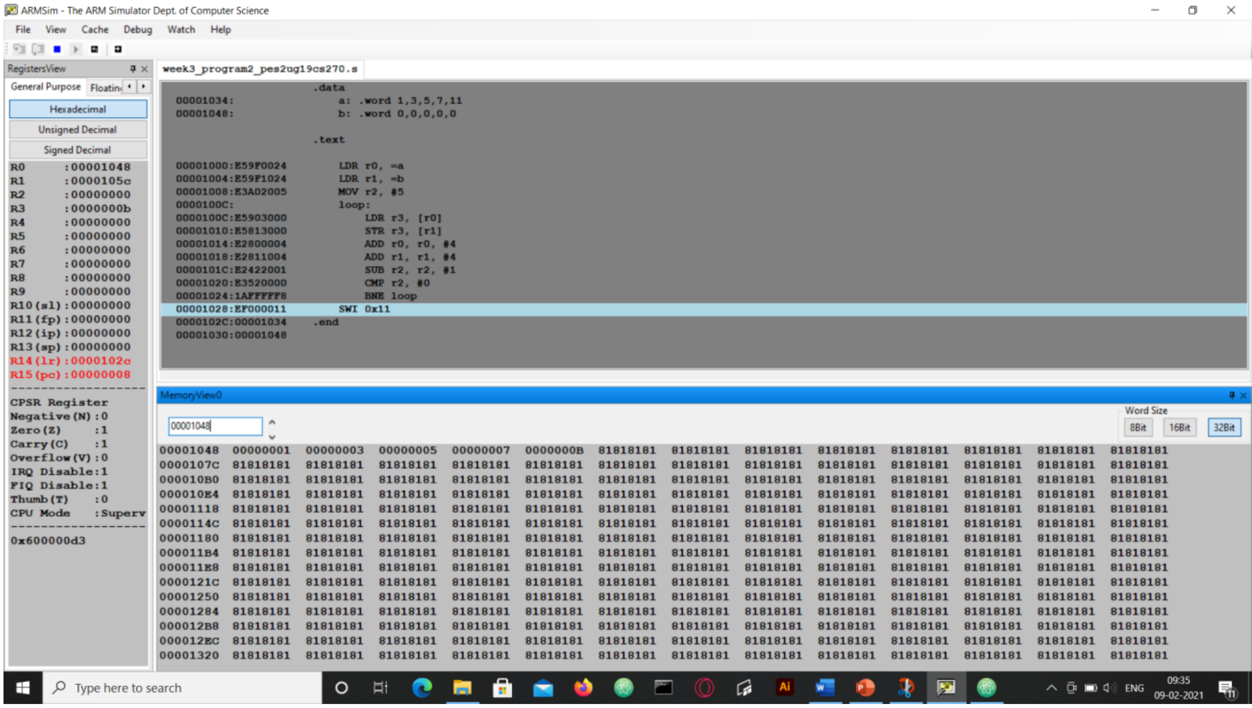
☑ Show Welcome Guide when opening Atom

```
.data
    a: .word 1,3,5,7,11
    b: .word 0,0,0,0,0

.text

    LDR r0, =a
    LDR r1, =b
    MOV r2, #5
    loop:
        LDR r3, [r0]
        STR r3, [r1]
        ADD r0, r0, #4
        ADD r1, r1, #4
        SUB r2, r2, #1
        CMP r2, #0
        BNE loop
    SWI 0x11
.end
```

## II. Output Screen Shot (One Example of your choice)



```
R0       :00001048
R1       :0000105c
R2       :00000000
R3       :0000000b
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :0000102c
R15(pc)  :00000008
-------------------
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   :Superv
-------------------
0x600000d3
```

```
                        .data
00001034:                   a: .word 1,3,5,7,11
00001048:                   b: .word 0,0,0,0,0

                        .text

00001000:E59F0024           LDR r0, =a
00001004:E59F1024           LDR r1, =b
00001008:E3A02005           MOV r2, #5
0000100C:                   loop:
0000100C:E5903000               LDR r3, [r0]
00001010:E5813000               STR r3, [r1]
00001014:E2800004               ADD r0, r0, #4
00001018:E2811004               ADD r1, r1, #4
0000101C:E2422001               SUB r2, r2, #1
00001020:E3520000               CMP r2, #0
00001024:1AFFFFF8               BNE loop
00001028:EF000011           SWI 0x11
0000102C:00001034       .end
00001030:00001048
```

```
00001048
00001048  00000001  00000003  00000005  00000007  0000000B  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000107C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010E4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001118  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000114C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001180  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000011B4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000011E8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000121C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001250  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001284  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000012B8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000012EC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001320  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

## III. Output Table for the program

| **.data**<br>**a: .word 1, 3, 5, 7,11**<br>**b: .word 0, 0, 0, 0,0** | |
|---|---|
| 1st Iteration | a: .word 01, 03, 05, 07,0B<br>b: .word 01, 0, 0, 0,0 |
| 2nd Iteration | a: .word 01, 03, 05, 07,0B<br>b: .word 01, 03, 0, 0,0 |
| 3rd Iteration | a: .word 01, 03, 05, 07,0B<br>b: .word 01, 03, 05, 0,0 |
| 4th Iteration | a: .word 01, 03, 05, 07,0B<br>b: .word 01, 03, 05, 07,0 |
| 5th Iteration | a: .word 01, 03, 05, 07,0B<br>b: .word 01, 03, 05, 07,0B |

## Microprocessor and Computer Architecture Laboratory

## UE19CS256

## 4th Semester, Academic Year 2020-21

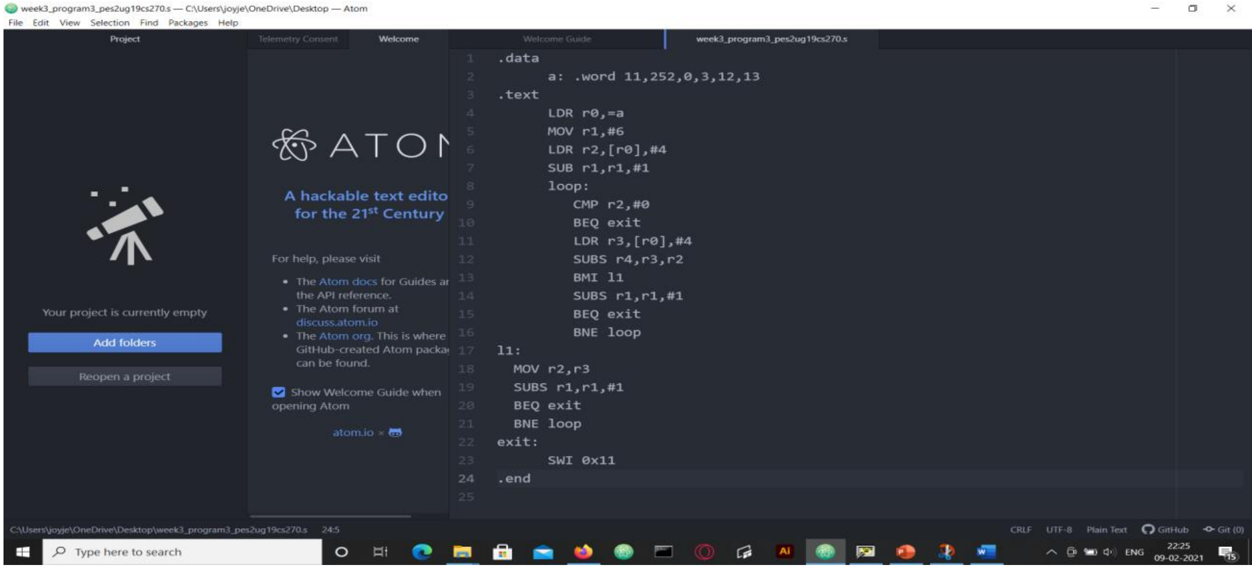
Date:

| | | |
|---|---|---|
| Name: OP JOY JEFERSON | SRN:PES2UG19CS270 | Section:E |

Week#____3_______          Program Number: ____3__

### Write an ALP to find smallest number in an array of n 32 bit numbers

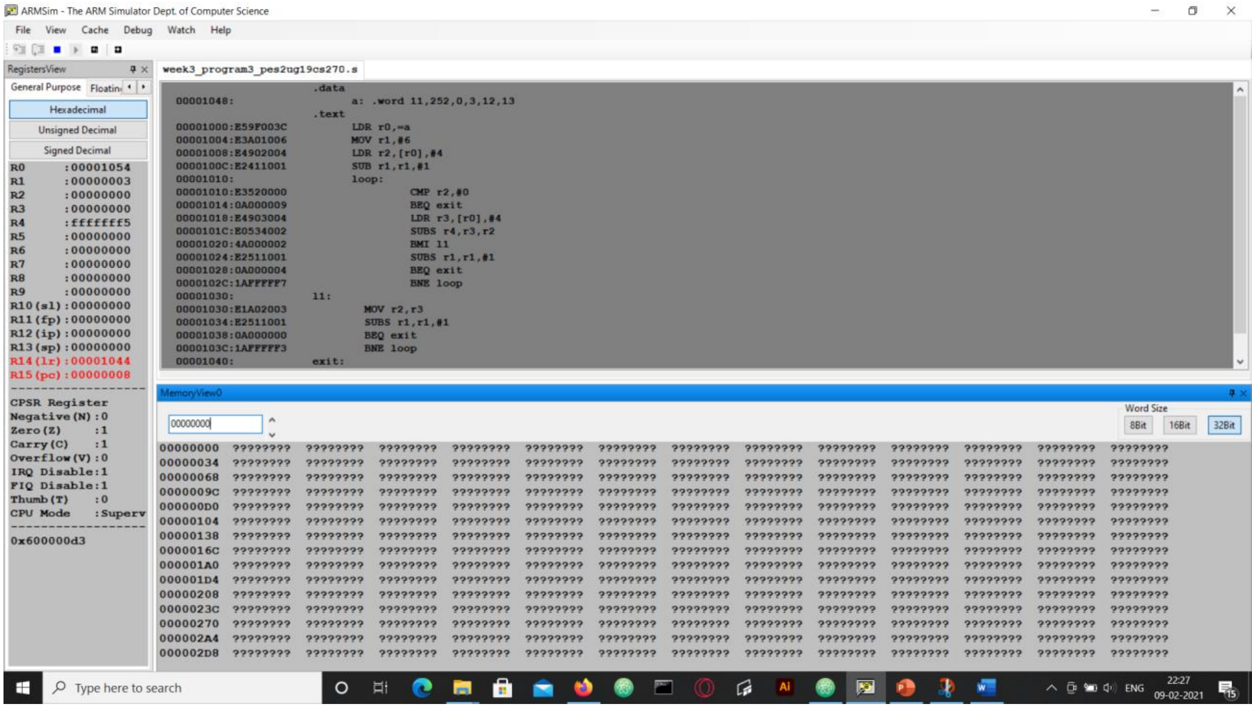
I. ARM Assembly Code for the program.



week3_program3_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Telemetry Consent
Welcome
Welcome Guide
week3_program3_pes2ug19cs270.s
Your project is currently empty
Add folders
Reopen a project
ATOM
A hackable text edito
for the 21st Century
For help, please visit
The Atom docs for Guides ar the API reference.
The Atom forum at discuss.atom.io
The Atom org. This is where GitHub-created Atom packa can be found.
☑ Show Welcome Guide when opening Atom
atom.io ×
C:\Users\joyje\OneDrive\Desktop\week3_program3_pes2ug19cs270.s 24:5
CRLF UTF-8 Plain Text GitHub Git (0)
Type here to search
ENG 22:25 09-02-2021

```
.data
      a: .word 11,252,0,3,12,13
.text
      LDR r0,=a
      MOV r1,#6
      LDR r2,[r0],#4
      SUB r1,r1,#1
      loop:
         CMP r2,#0
         BEQ exit
         LDR r3,[r0],#4
         SUBS r4,r3,r2
         BMI l1
         SUBS r1,r1,#1
         BEQ exit
         BNE loop
l1:
  MOV r2,r3
  SUBS r1,r1,#1
  BEQ exit
  BNE loop
exit:
      SWI 0x11
.end
```

II. Output Screen Shot (One Example of your choice)



```
R0       :00001054
R1       :00000003
R2       :00000000
R3       :00000000
R4       :fffffff5
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :00001044
R15(pc)  :00000008
------------------
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   :Superv
------------------
0x600000d3
```

```
                        .data
00001048:                   a: .word 11,252,0,3,12,13
                        .text
00001000:E59F003C           LDR r0,=a
00001004:E3A01006           MOV r1,#6
00001008:E4902004           LDR r2,[r0],#4
0000100C:E2411001           SUB r1,r1,#1
00001010:                   loop:
00001010:E3520000                   CMP r2,#0
00001014:0A000009                   BEQ exit
00001018:E4903004                   LDR r3,[r0],#4
0000101C:E0534002                   SUBS r4,r3,r2
00001020:4A000002                   BMI l1
00001024:E2511001                   SUBS r1,r1,#1
00001028:0A000004                   BEQ exit
0000102C:1AFFFFF7                   BNE loop
00001030:               l1:
00001030:E1A02003             MOV r2,r3
00001034:E2511001             SUBS r1,r1,#1
00001038:0A000000             BEQ exit
0000103C:1AFFFFF3             BNE loop
00001040:               exit:
```

```
00000000
00000000  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000034  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000068  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
0000009C  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
000000D0  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000104  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000138  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
0000016C  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
000001A0  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
000001D4  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000208  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
0000023C  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
00000270  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
000002A4  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
000002D8  ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ???????? ????????
```

## III. Output Table for the program

| a: .word 11,252,0,3,12,13 | |
|---|---|
| 1st Iteration | R2=11,R3=252 (R3>R2) |
| 2nd Iteration | R2=11, R3=0 (R3<R2) |
| 3rd Iteration | R2=0 |
| 4th Iteration | - |
| 5th Iteration | - |
| Smallest number is present in R2 | |

**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

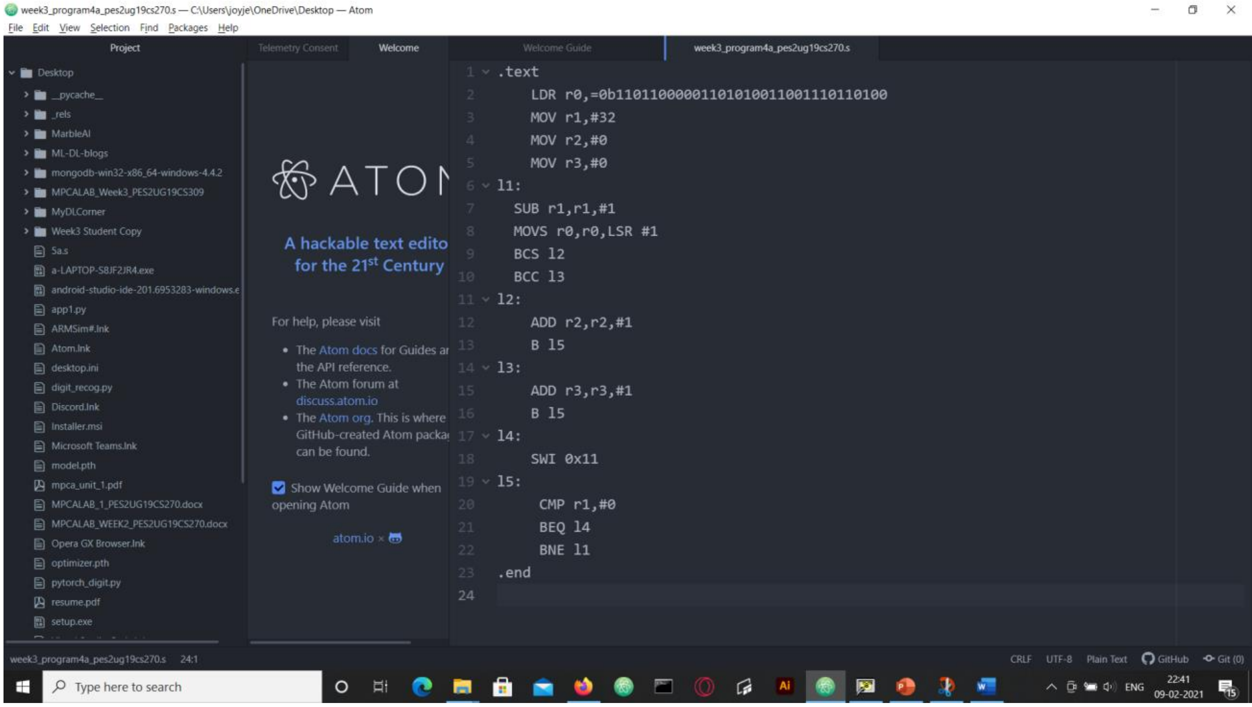
Date:9/02/2021

| | | |
|---|---|---|
| Name: OP JOY JEFFERSON | SRN:PES2UG19CS270 | Section:E |

Week#____3_______ Program Number: ____4a_

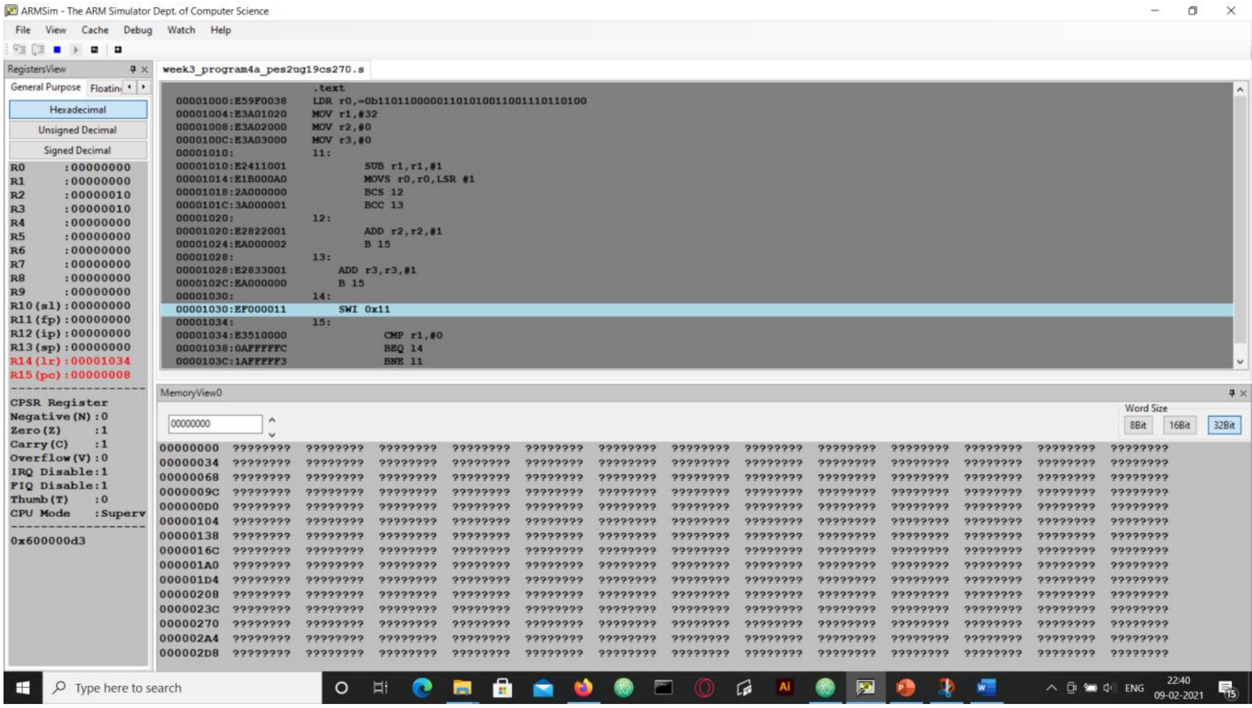
**Write an ALP to count the number of 1's and 0's in a given 32 bit number.**

I. ARM Assembly Code for the program.



```
.text
    LDR r0,=0b11011000001101010011001110110100
    MOV r1,#32
    MOV r2,#0
    MOV r3,#0
l1:
  SUB r1,r1,#1
  MOVS r0,r0,LSR #1
  BCS l2
  BCC l3
l2:
    ADD r2,r2,#1
    B l5
l3:
    ADD r3,r3,#1
    B l5
l4:
    SWI 0x11
l5:
     CMP r1,#0
     BEQ l4
     BNE l1
.end
```

## II. Output Screen Shot (One Example of your choice)



week3_program4a_pes2ug19cs270.s

```
                      .text
00001000:E59F0038     LDR r0,=0b11011000001101010011001110110100
00001004:E3A01020     MOV r1,#32
00001008:E3A02000     MOV r2,#0
0000100C:E3A03000     MOV r3,#0
00001010:             l1:
00001010:E2411001             SUB r1,r1,#1
00001014:E1B000A0             MOVS r0,r0,LSR #1
00001018:2A000000             BCS l2
0000101C:3A000001             BCC l3
00001020:             l2:
00001020:E2822001             ADD r2,r2,#1
00001024:EA000002             B l5
00001028:             l3:
00001028:E2833001         ADD r3,r3,#1
0000102C:EA000000         B l5
00001030:             l4:
00001030:EF000011         SWI 0x11
00001034:             l5:
00001034:E3510000                CMP r1,#0
00001038:0AFFFFFC                BEQ l4
0000103C:1AFFFFF3                BNE l1
```

```
R0       :00000000
R1       :00000000
R2       :00000010
R3       :00000010
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :00001034
R15(pc)  :00000008
-----------------
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   :Superv
-----------------
0x600000d3
```

MemoryView0

```
00000000  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
00000034  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
00000068  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
0000009C  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
000000D0  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
00000104  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
00000138  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
0000016C  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
000001A0  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
000001D4  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
00000208  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
0000023C  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
00000270  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
000002A4  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
000002D8  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????  ????????
```

## III. Output Table for the program

| r0, =0b11011000001101010011001110110100 | | |
|---|---|---|
| r1 | 32 | |
| r2 | After execution | 16<br>(=10 in hex) |
| r3 | After execution | 16<br>(=10 in hex) |

# Microprocessor and Computer Architecture Laboratory

# UE19CS256

# 4th Semester, Academic Year 2020-21

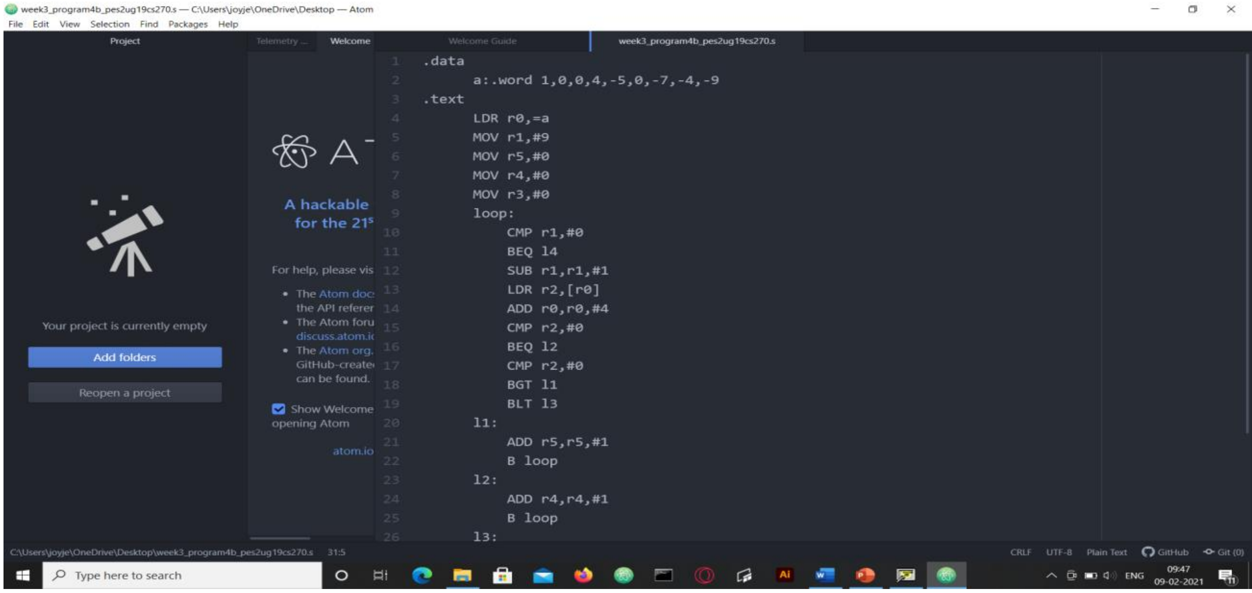
Date:

| | | |
|---|---|---|
| Name:OP JOY JEFFERSON | SRN:PES2UG19CS270 | Section:E |

Week#____3_______ Program Number: ____4b_

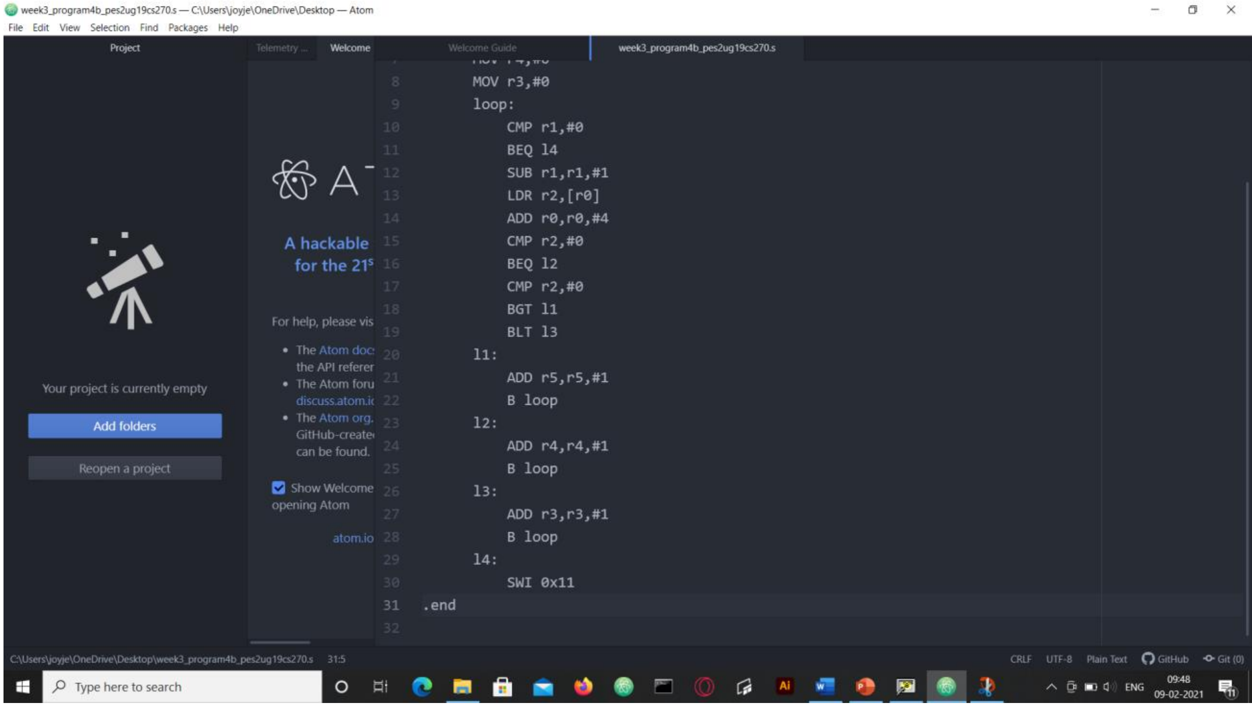
## Write an ALP to find the number of zeroes, positive and negative numbers in a given array

I. ARM Assemby Code for the program.



week3_program4b_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Telemetry ...
Welcome
Welcome Guide
week3_program4b_pes2ug19cs270.s
A hackable
for the 21s
For help, please vis
The Atom doc
the API referer
The Atom foru
discuss.atom.ic
The Atom org.
GitHub-create
can be found.
Show Welcome
opening Atom
atom.io
Your project is currently empty
Add folders
Reopen a project
C:\Users\joyje\OneDrive\Desktop\week3_program4b_pes2ug19cs270.s 31:5
CRLF UTF-8 Plain Text GitHub Git (0)
Type here to search
ENG 09:47 09-02-2021

```
.data
      a:.word 1,0,0,4,-5,0,-7,-4,-9
.text
      LDR r0,=a
      MOV r1,#9
      MOV r5,#0
      MOV r4,#0
      MOV r3,#0
      loop:
          CMP r1,#0
          BEQ l4
          SUB r1,r1,#1
          LDR r2,[r0]
          ADD r0,r0,#4
          CMP r2,#0
          BEQ l2
          CMP r2,#0
          BGT l1
          BLT l3
      l1:
          ADD r5,r5,#1
          B loop
      l2:
          ADD r4,r4,#1
          B loop
      l3:
```

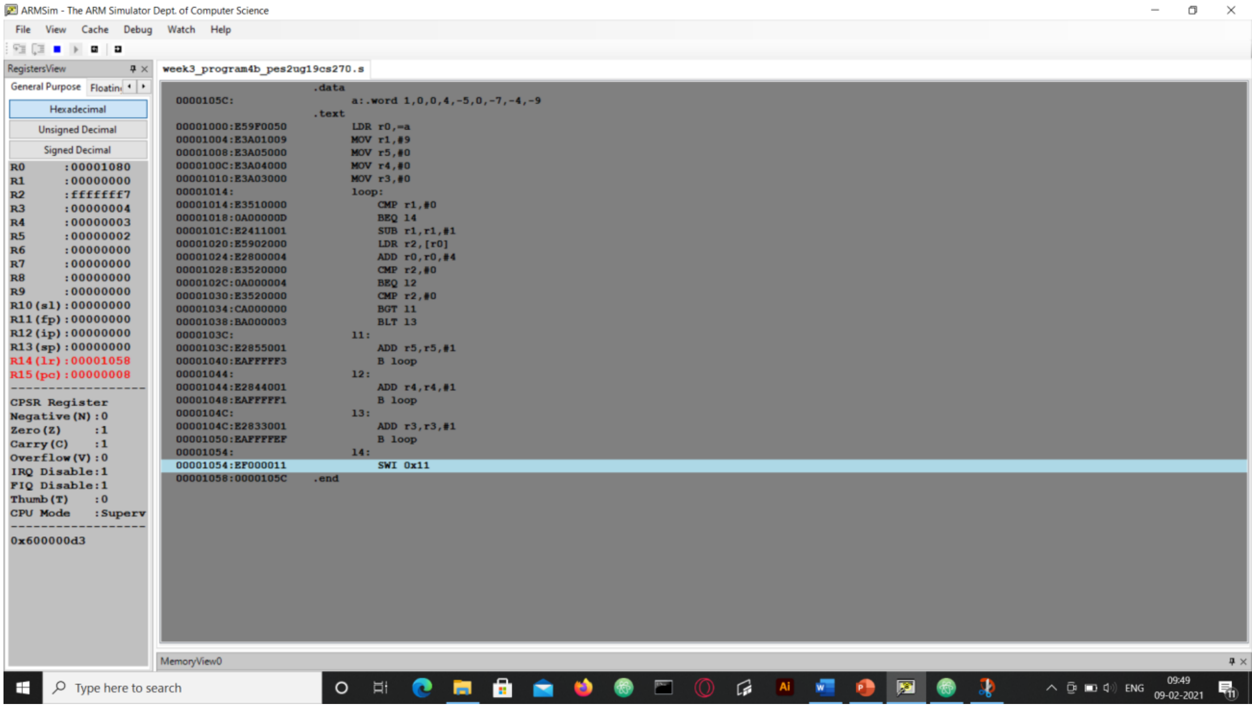


week3_program4b_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom

```
    MOV r3,#0
    loop:
        CMP r1,#0
        BEQ l4
        SUB r1,r1,#1
        LDR r2,[r0]
        ADD r0,r0,#4
        CMP r2,#0
        BEQ l2
        CMP r2,#0
        BGT l1
        BLT l3
    l1:
        ADD r5,r5,#1
        B loop
    l2:
        ADD r4,r4,#1
        B loop
    l3:
        ADD r3,r3,#1
        B loop
    l4:
        SWI 0x11
.end
```

C:\Users\joyje\OneDrive\Desktop\week3_program4b_pes2ug19cs270.s 31:5

## II. Output Screen Shot (One Example of your choice)



ARMSim - The ARM Simulator Dept. of Computer Science

week3_program4b_pes2ug19cs270.s

```
R0       :00001080
R1       :00000000
R2       :fffffff7
R3       :00000004
R4       :00000003
R5       :00000002
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :00001058
R15(pc)  :00000008
------------------
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   :Superv
------------------
0x600000d3
```

```
                         .data
0000105C:                     a:.word 1,0,0,4,-5,0,-7,-4,-9
                         .text
00001000:E59F0050             LDR r0,=a
00001004:E3A01009             MOV r1,#9
00001008:E3A05000             MOV r5,#0
0000100C:E3A04000             MOV r4,#0
00001010:E3A03000             MOV r3,#0
00001014:                     loop:
00001014:E3510000                 CMP r1,#0
00001018:0A00000D                 BEQ l4
0000101C:E2411001                 SUB r1,r1,#1
00001020:E5902000                 LDR r2,[r0]
00001024:E2800004                 ADD r0,r0,#4
00001028:E3520000                 CMP r2,#0
0000102C:0A000004                 BEQ l2
00001030:E3520000                 CMP r2,#0
00001034:CA000000                 BGT l1
00001038:BA000003                 BLT l3
0000103C:                     l1:
0000103C:E2855001                 ADD r5,r5,#1
00001040:EAFFFFF3                 B loop
00001044:                     l2:
00001044:E2844001                 ADD r4,r4,#1
00001048:EAFFFFF1                 B loop
0000104C:                     l3:
0000104C:E2833001                 ADD r3,r3,#1
00001050:EAFFFFEF                 B loop
00001054:                     l4:
00001054:EF000011                 SWI 0x11
00001058:0000105C        .end
```

## III. Output Table for the program

| a:.word 1,0,0,4,-5,0,-7,-4,-9 | | |
|---|---|---|
| R3 | 4 | |
| R4 | 3 | |
| R5 | 2 | |

**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

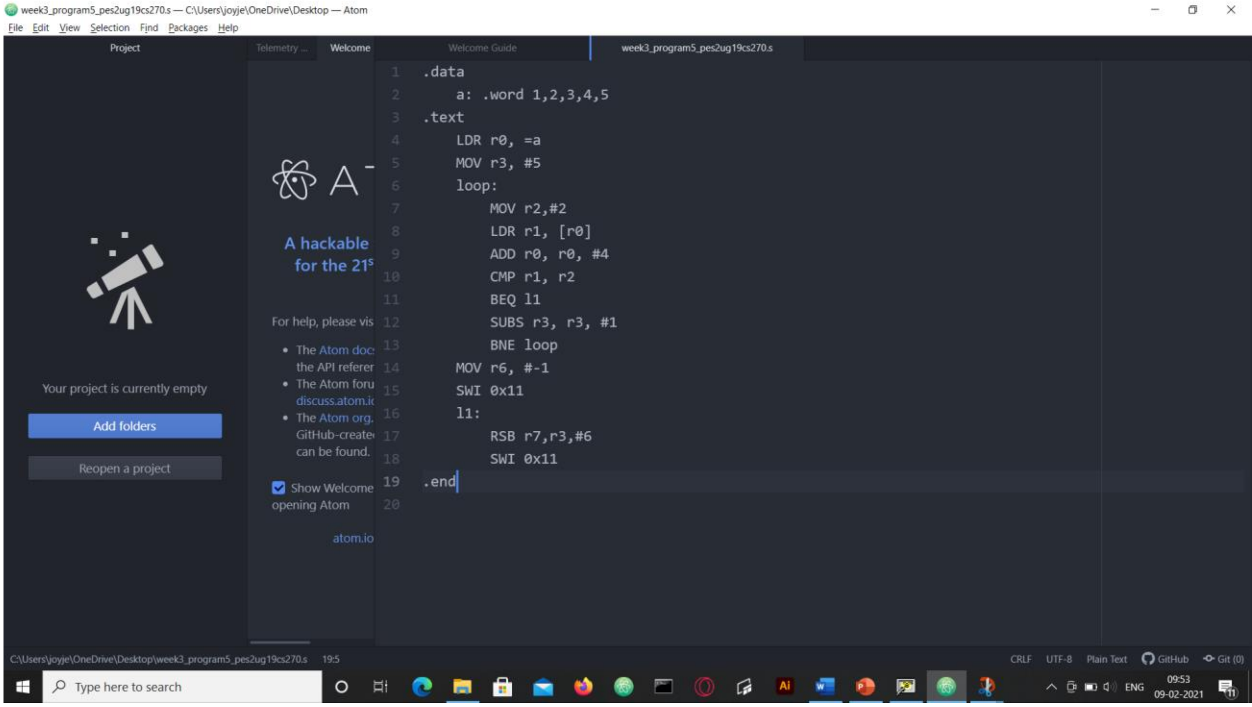
Date:

| | | |
|---|---|---|
| Name: OP JOY JEFFERSON | SRN:PES2UG19CS270 | Section:E |

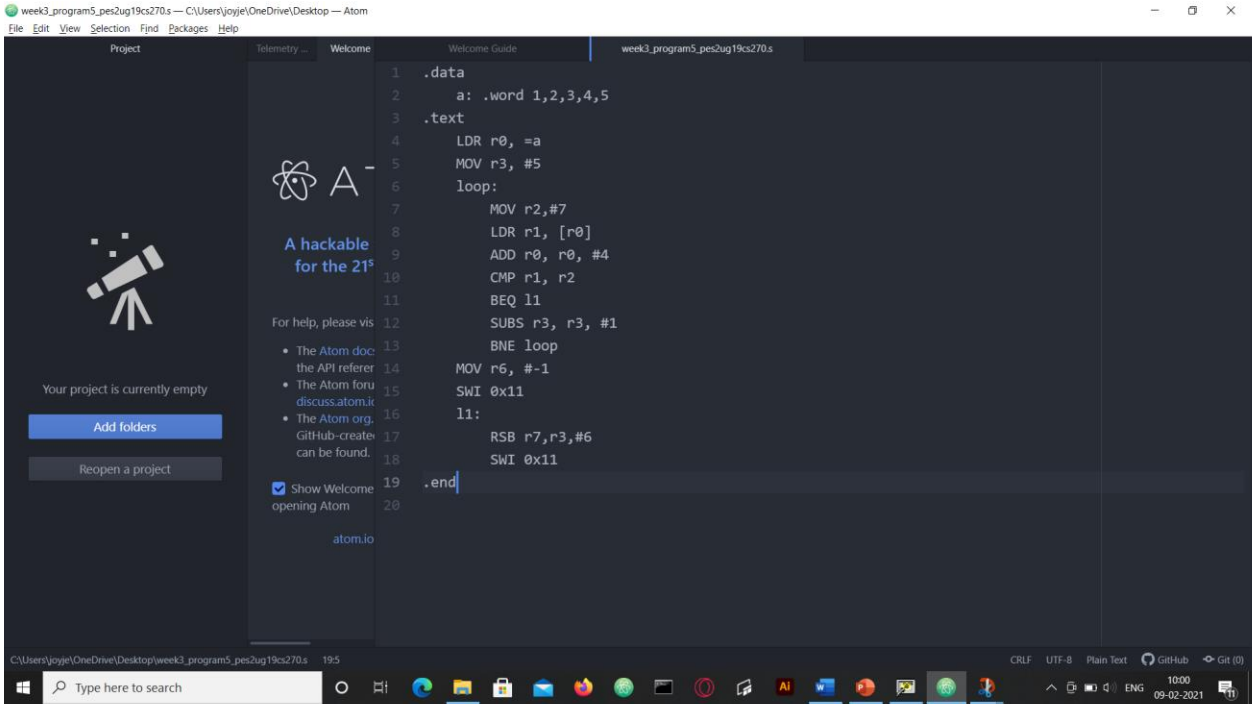
Week#____3_______ Program Number: ____5_

**Write an ALP to check whether a given number is present in array using Linear Search (Without SWI 0x02), if found move +1 to R6 and key position to R7 else move -1 to R6 (if number not found)**

I. ARM Assembly Code for the program.

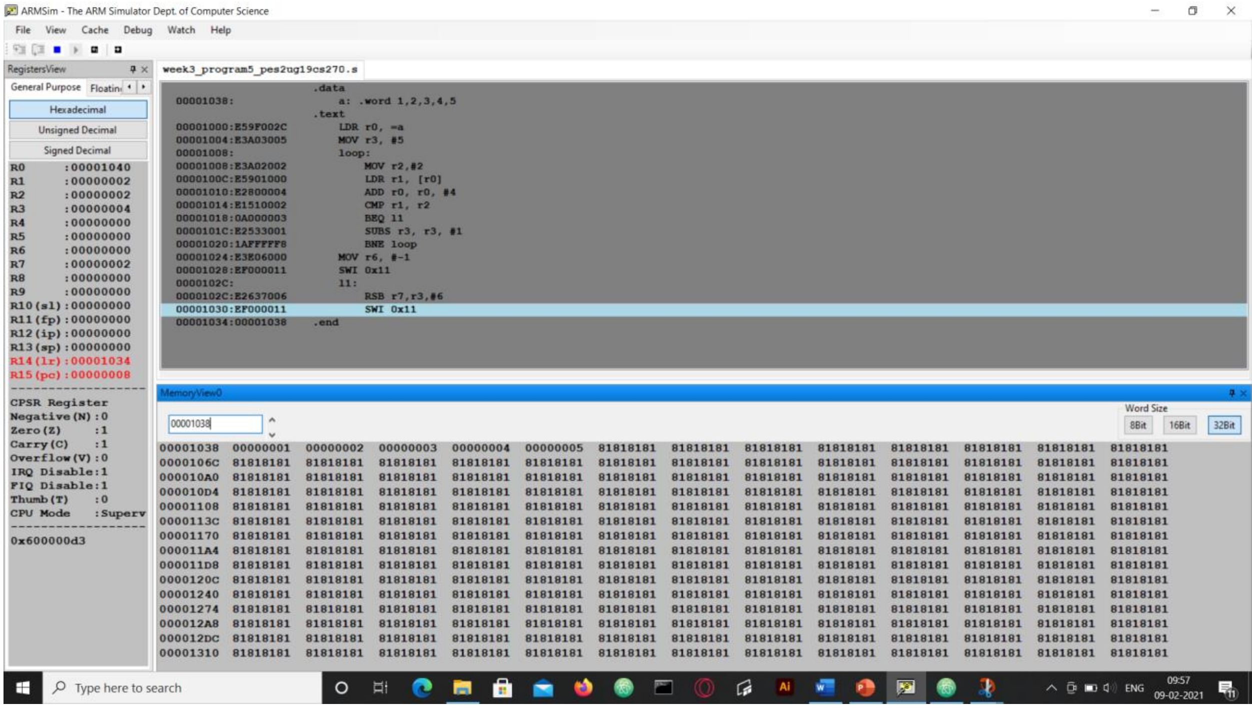

week3_program5_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Welcome
Welcome Guide
week3_program5_pes2ug19cs270.s
Your project is currently empty
Add folders
Reopen a project
C:\Users\joyje\OneDrive\Desktop\week3_program5_pes2ug19cs270.s 19:5
CRLF UTF-8 Plain Text GitHub Git (0)
Type here to search
09:53
09-02-2021

```
.data
    a: .word 1,2,3,4,5
.text
    LDR r0, =a
    MOV r3, #5
    loop:
        MOV r2,#2
        LDR r1, [r0]
        ADD r0, r0, #4
        CMP r1, r2
        BEQ l1
        SUBS r3, r3, #1
        BNE loop
    MOV r6, #-1
    SWI 0x11
    l1:
        RSB r7,r3,#6
        SWI 0x11
.end
```


week3_program5_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Welcome
Welcome Guide
week3_program5_pes2ug19cs270.s
Your project is currently empty
Add folders
Reopen a project
C:\Users\joyje\OneDrive\Desktop\week3_program5_pes2ug19cs270.s 19:5
CRLF UTF-8 Plain Text GitHub Git (0)
Type here to search
10:00
09-02-2021

```
.data
    a: .word 1,2,3,4,5
.text
    LDR r0, =a
    MOV r3, #5
    loop:
        MOV r2,#7
        LDR r1, [r0]
        ADD r0, r0, #4
        CMP r1, r2
        BEQ l1
        SUBS r3, r3, #1
        BNE loop
    MOV r6, #-1
    SWI 0x11
    l1:
        RSB r7,r3,#6
        SWI 0x11
.end
```

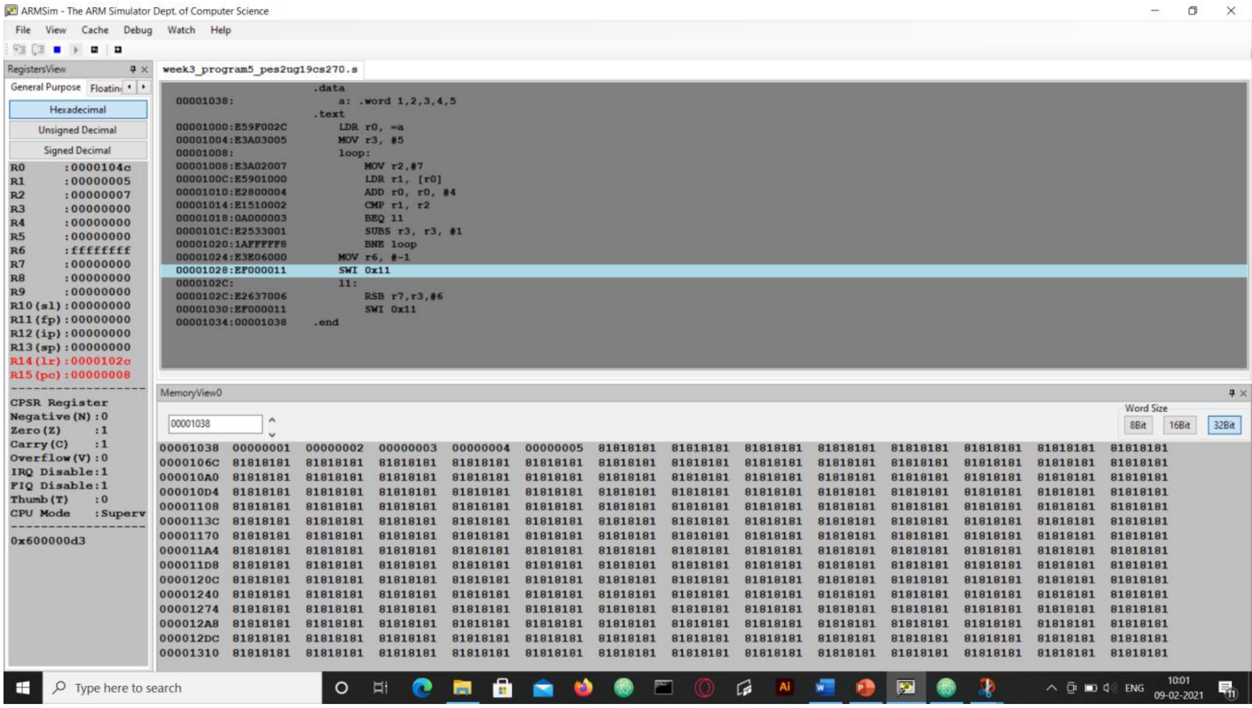
## II. Output Screen Shot (One Example of your choice)



```
                        .data
00001038:                   a: .word 1,2,3,4,5
                        .text
00001000:E59F002C           LDR r0, =a
00001004:E3A03005           MOV r3, #5
00001008:                   loop:
00001008:E3A02002               MOV r2,#2
0000100C:E5901000               LDR r1, [r0]
00001010:E2800004               ADD r0, r0, #4
00001014:E1510002               CMP r1, r2
00001018:0A000003               BEQ l1
0000101C:E2533001               SUBS r3, r3, #1
00001020:1AFFFFF8               BNE loop
00001024:E3E06000           MOV r6, #-1
00001028:EF000011           SWI 0x11
0000102C:                   l1:
0000102C:E2637006               RSB r7,r3,#6
00001030:EF000011               SWI 0x11
00001034:00001038       .end
```

```
R0       :00001040
R1       :00000002
R2       :00000002
R3       :00000004
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000002
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :00001034
R15(pc)  :00000008
------------------
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    :Superv
------------------
0x600000d3
```

```
00001038
00001038  00000001  00000002  00000003  00000004  00000005  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000106C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010A0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010D4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001108  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000113C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001170  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000011A4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000011D8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000120C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001240  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001274  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000012A8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000012DC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001310  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```



```
                        .data
00001038:                   a: .word 1,2,3,4,5
                        .text
00001000:E59F002C           LDR r0, =a
00001004:E3A03005           MOV r3, #5
00001008:                   loop:
00001008:E3A02007               MOV r2,#7
0000100C:E5901000               LDR r1, [r0]
00001010:E2800004               ADD r0, r0, #4
00001014:E1510002               CMP r1, r2
00001018:0A000003               BEQ l1
0000101C:E2533001               SUBS r3, r3, #1
00001020:1AFFFFF8               BNE loop
00001024:E3E06000           MOV r6, #-1
00001028:EF000011           SWI 0x11
0000102C:                   l1:
0000102C:E2637006               RSB r7,r3,#6
00001030:EF000011               SWI 0x11
00001034:00001038       .end
```

```
R0       :0000104c
R1       :00000005
R2       :00000007
R3       :00000000
R4       :00000000
R5       :00000000
R6       :ffffffff
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00000000
R14(lr)  :0000102c
R15(pc)  :00000008
------------------
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    :Superv
------------------
0x600000d3
```

```
00001038
00001038  00000001  00000002  00000003  00000004  00000005  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000106C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010A0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010D4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001108  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000113C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001170  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000011A4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000011D8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000120C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001240  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001274  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000012A8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000012DC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001310  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

## III. Output Table for the program

| | | |
|---|---|---|
| | | HEX value |
| A:.WORD 1,2,3,4,5 | | |
| R2 | KEY =2 | 02 |
| R3 | COUNT =5 | |
| R0 | Address of A | 00001038 |
| R3 | After Execution =4 | Position of key element =(n+1)-R3=6-4=2 |

| | | |
|---|---|---|
| | | HEX value |
| A:.WORD 1,2,3,4,5 | | |
| R2 | KEY =7 | 07 |
| R3 | COUNT =5 | |
| R0 | Address of A | 00001038 |
| R3 | After Execution =0 | Position of key element R6=-1 |

## Microprocessor and Computer Architecture Laboratory

## UE19CS256

## 4th Semester, Academic Year 2020-21

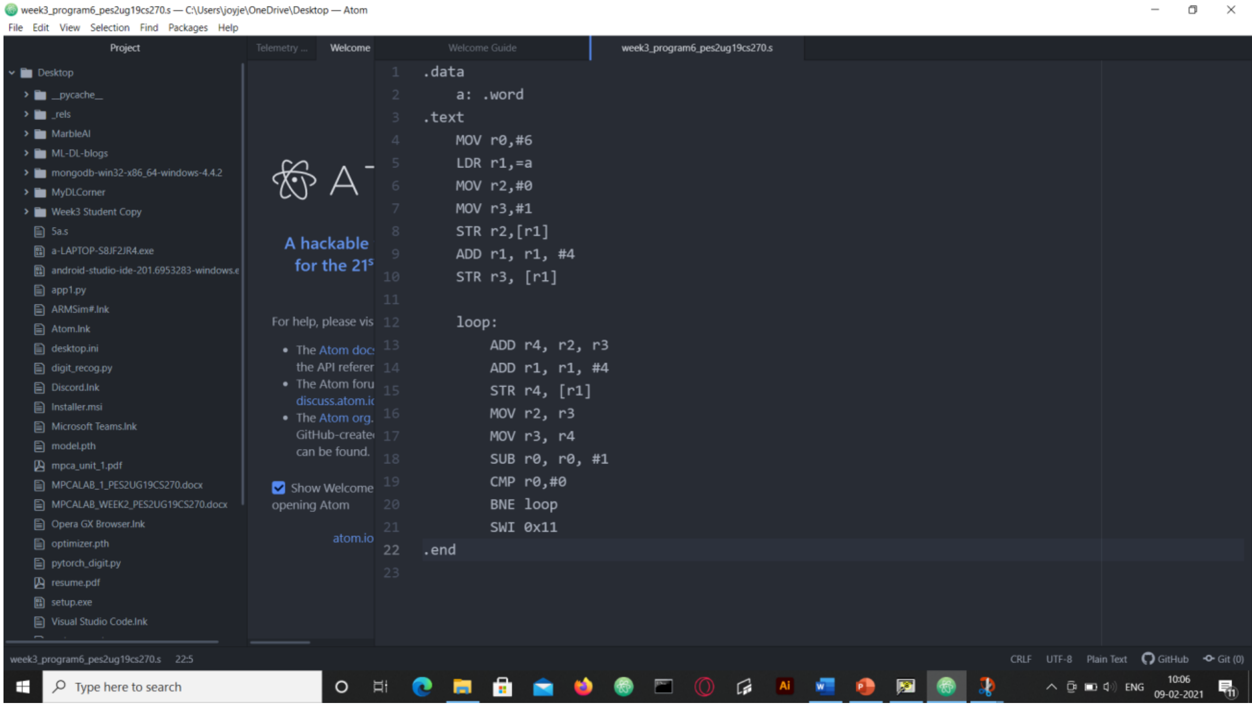
Date:

| | | |
|---|---|---|
| Name: OP JOY JEFFERSON | SRN:PES2UG19CS270 | Section:E |

Week#____3_______ Program Number: ____6_

# Write an ALP to generate Fibonacci Series and store them in an array

## I. ARM Assembly Code for the program.

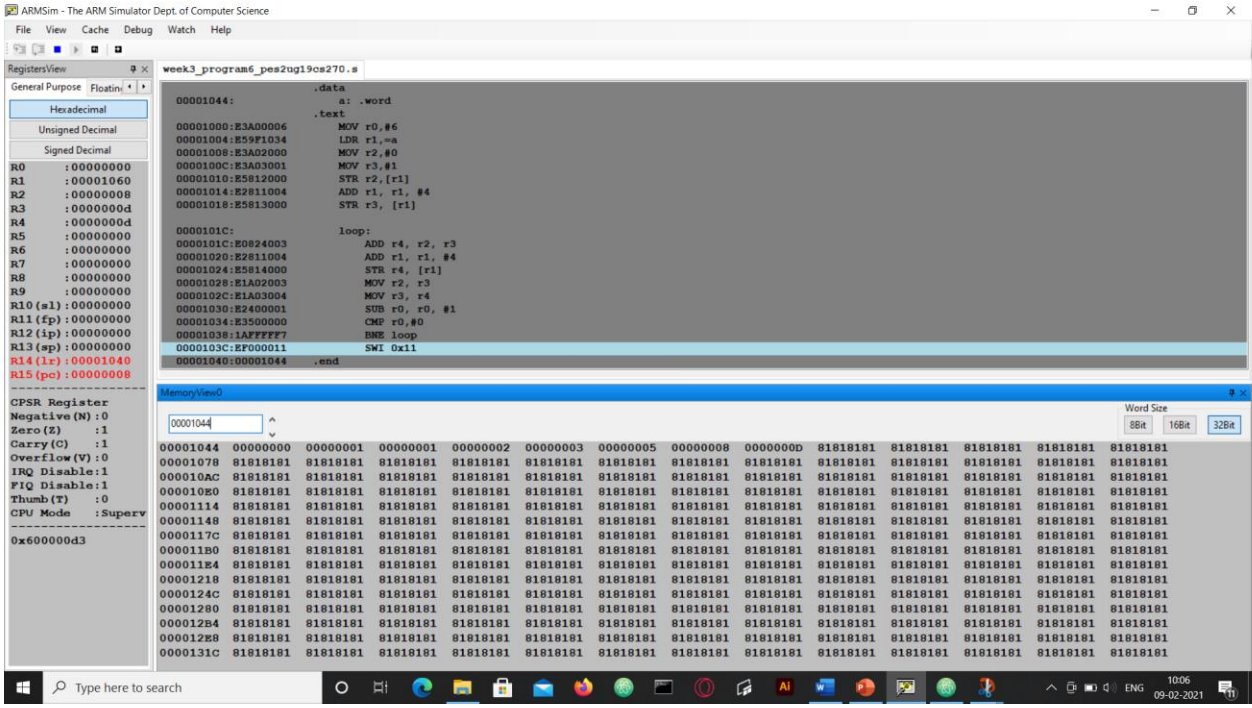


week3_program6_pes2ug19cs270.s — C:\Users\joyje\OneDrive\Desktop — Atom
File Edit View Selection Find Packages Help
Project
Telemetry ...
Welcome
Welcome Guide
week3_program6_pes2ug19cs270.s
week3_program6_pes2ug19cs270.s 22:5
CRLF UTF-8 Plain Text GitHub Git (0)
Type here to search
10:06
09-02-2021

```
.data
    a: .word
.text
    MOV r0,#6
    LDR r1,=a
    MOV r2,#0
    MOV r3,#1
    STR r2,[r1]
    ADD r1, r1, #4
    STR r3, [r1]

    loop:
        ADD r4, r2, r3
        ADD r1, r1, #4
        STR r4, [r1]
        MOV r2, r3
        MOV r3, r4
        SUB r0, r0, #1
        CMP r0,#0
        BNE loop
        SWI 0x11
.end
```

## II. Output Screen Shot (One Example of your choice)



```
R0      :00000000
R1      :00001060
R2      :00000008
R3      :0000000d
R4      :0000000d
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00000000
R14(lr) :00001040
R15(pc) :00000008
--------------------
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   :Superv
--------------------
0x600000d3
```

```
                          .data
00001044:                     a: .word
                          .text
00001000:E3A00006             MOV r0,#6
00001004:E59F1034             LDR r1,=a
00001008:E3A02000             MOV r2,#0
0000100C:E3A03001             MOV r3,#1
00001010:E5812000             STR r2,[r1]
00001014:E2811004             ADD r1, r1, #4
00001018:E5813000             STR r3, [r1]

0000101C:                     loop:
0000101C:E0824003                 ADD r4, r2, r3
00001020:E2811004                 ADD r1, r1, #4
00001024:E5814000                 STR r4, [r1]
00001028:E1A02003                 MOV r2, r3
0000102C:E1A03004                 MOV r3, r4
00001030:E2400001                 SUB r0, r0, #1
00001034:E3500000                 CMP r0,#0
00001038:1AFFFFF7                 BNE loop
0000103C:EF000011                 SWI 0x11
00001040:00001044         .end
```

```
00001044
00001044 00000000 00000001 00000001 00000002 00000003 00000005 00000008 0000000D 81818181 81818181 81818181 81818181 81818181 81818181
00001078 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001114 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001148 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000117C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000011B0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000011E4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001218 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000124C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001280 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000012B4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000012E8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000131C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

## III. Output Table for the program

| | | |
|---|---|---|
| R0 | Fibonacci Count | 6 |
| R1 | Address of A | 00001044 |
| R2 | Initially 0 | |
| R3 | Initially 1 | |
| R4 | 1st Iteration | 0+1=1 |
| R4 | 2nd Iteration | 1+1=2 |
| R4 | 3rd Iteration | 2+1=3 |
| R4 | 4th Iteration | 3+2=5 |
| R4 | 5th Iteration | 5+3=8 |
| R4 | 6th Iteration | 8+5=13 =0D |

## Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature:OP JOY JEFFFERSON

Name:OP Joy Jefferson

SRN:PES2UG19CS270

Section: E

Date:9/02/2021