ARM JOURNAL

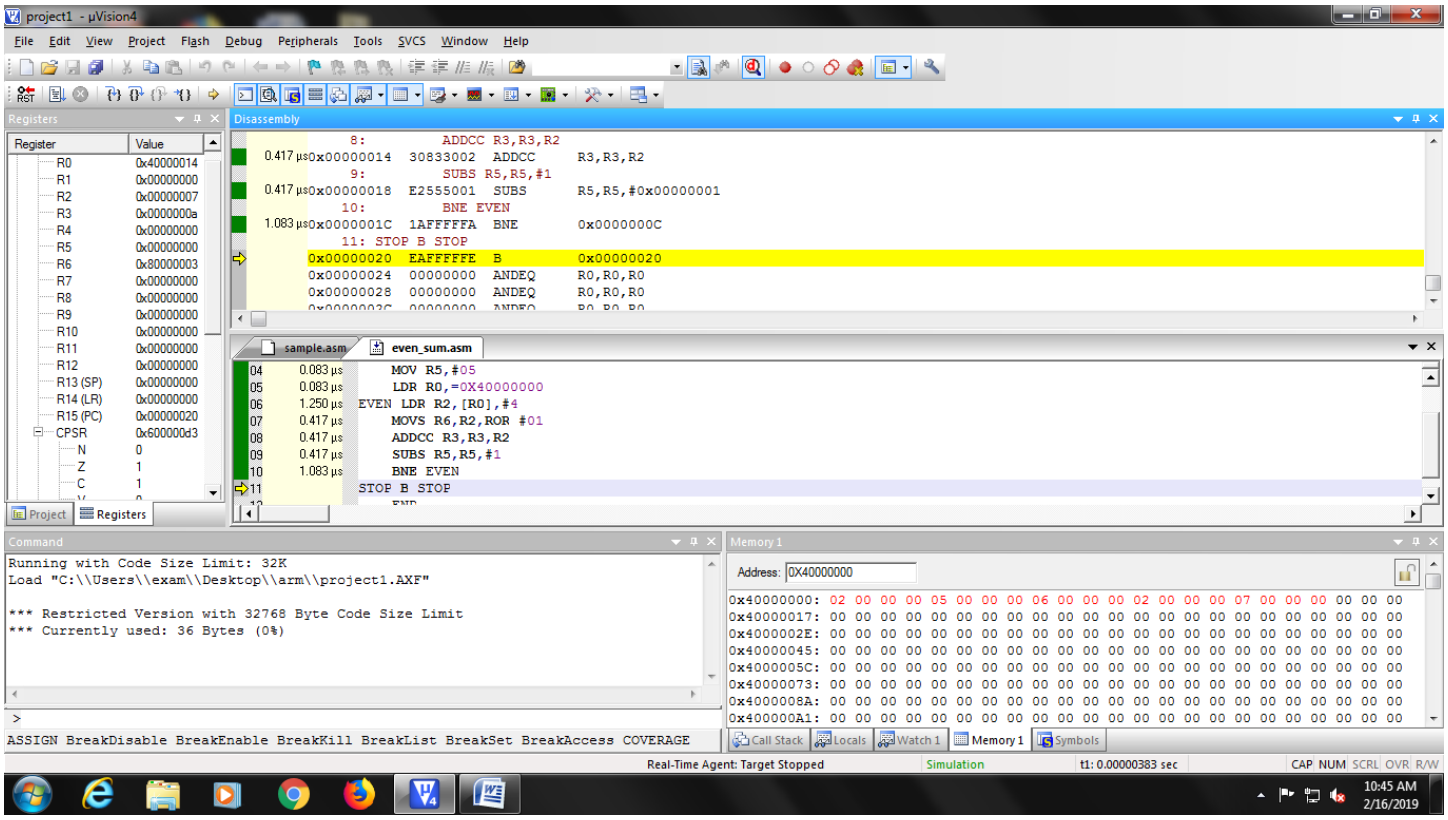
ARM7TDMI

PAVAN C E
01FE17BEC114
DIV:B(BATCH B2)
ROLL NO:139

//ASM TO FIND SUM OF EVEN NUMBERS IN A GIVEN ARRAY OF 5 ELEMENTS

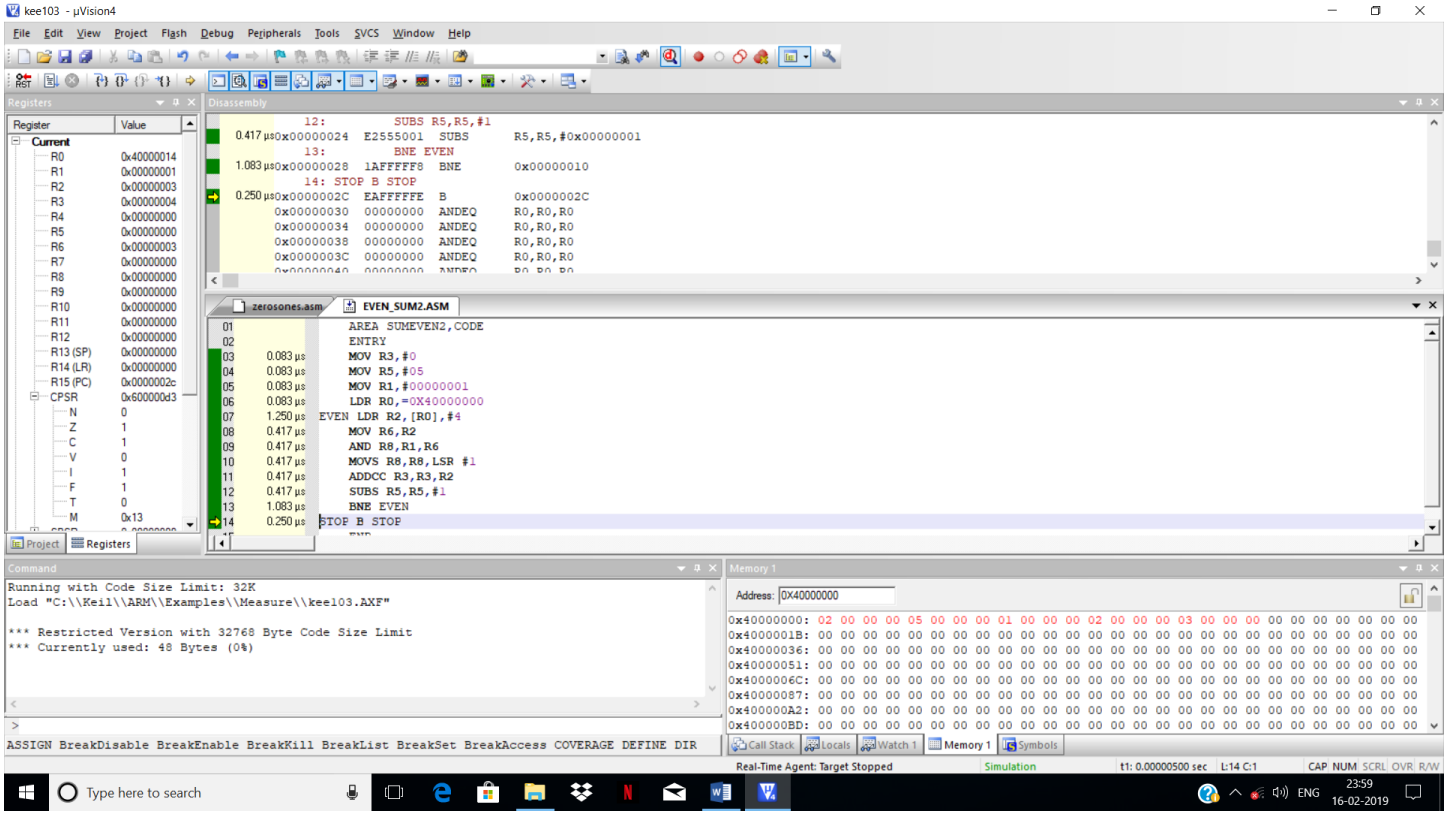
CODE 1:

```
        AREA SUMEVEN,CODE
        ENTRY
        MOV R3,#0
        MOV R5,#05
        LDR R0,=0X40000000
EVEN LDR R2,[R0],#4
        MOVS R6,R2,ROR #01
        ADDCC R3,R3,R2
        SUBS R5,R5,#1
        BNE EVEN
STOP B STOP
        END
```



CODE 2:

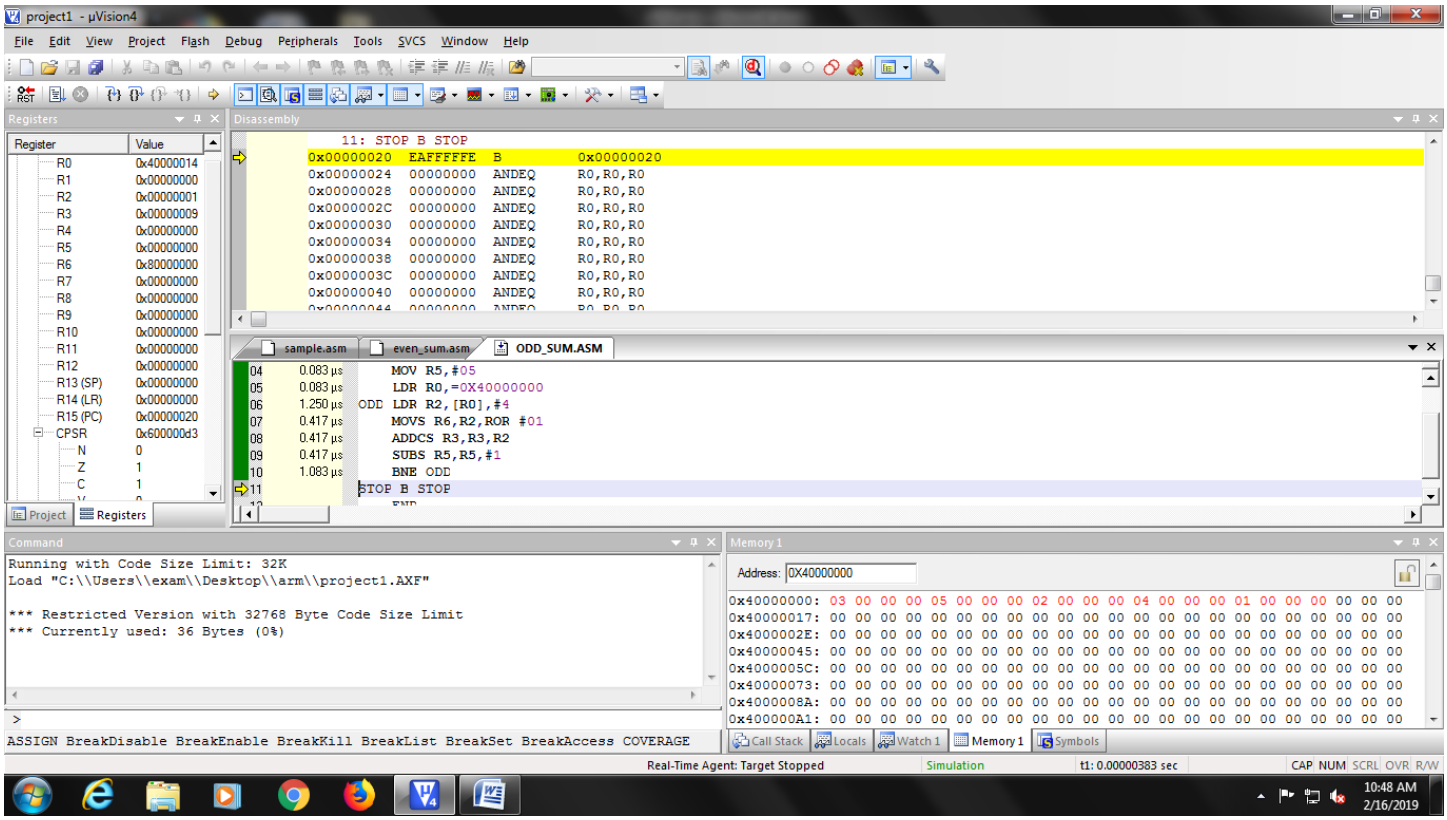
```
AREA SUMEVEN2,CODE
        ENTRY
        MOV R3,#0
        MOV R5,#05
        MOV R1,#00000001
        LDR R0,=0X40000000
EVEN LDR R2,[R0],#4
        MOV R6,R2
        AND R8,R1,R6
        MOVS R8,R8,LSR #1
        ADDCC R3,R3,R2
        SUBS R5,R5,#1
        BNE EVEN
STOP B STOP
        END
```



//ASM TO FIND SUM OF ODD NUMBERS IN A GIVEN ARRAY OF 5 ELEMENTS

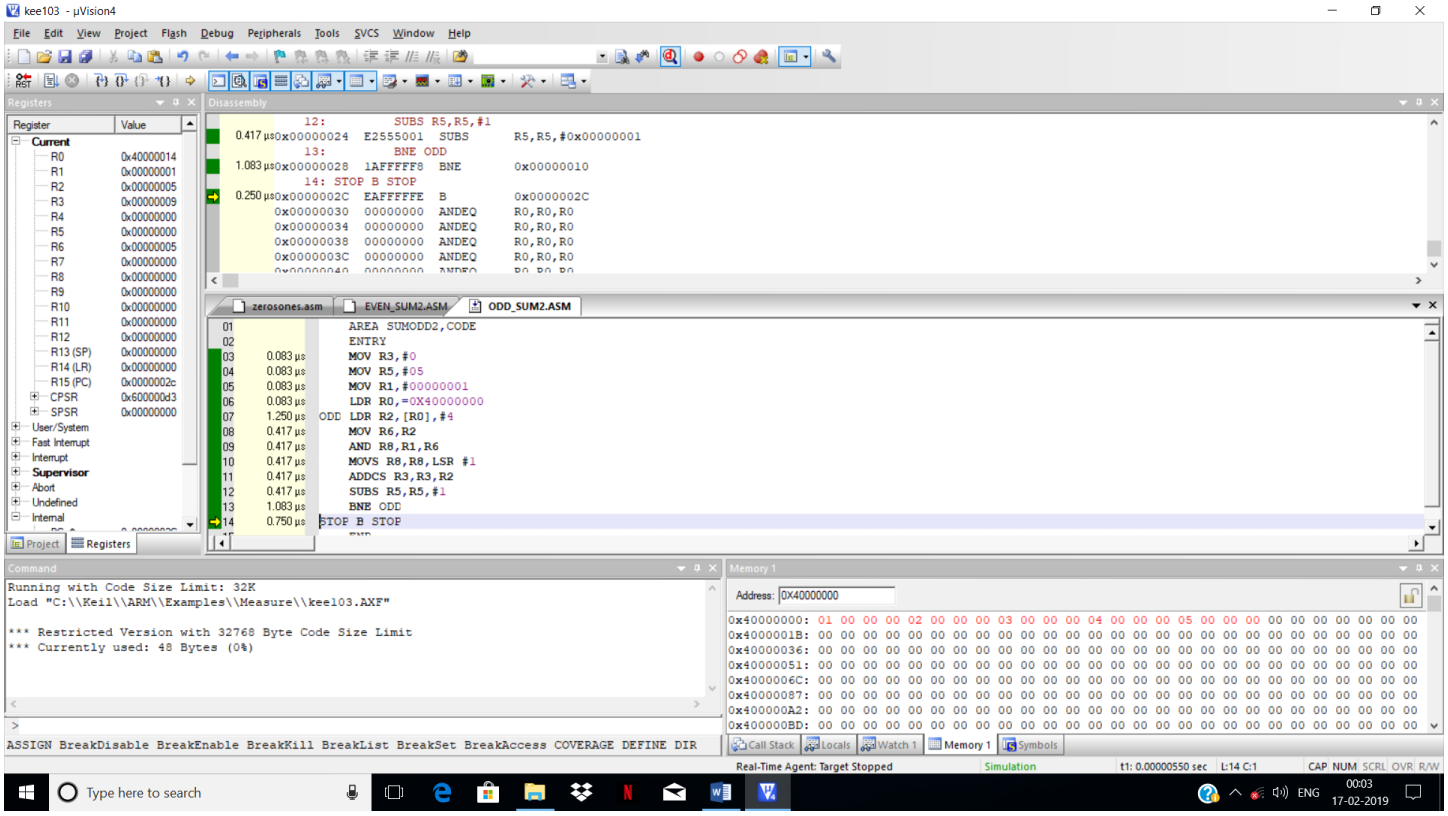
CODE 1:

```
AREA SUMODD,CODE
        ENTRY
        MOV R3,#0
        MOV R5,#05
        LDR R0,=0X40000000
ODD LDR R2,[R0],#4
        MOVS R6,R2,ROR #01
        ADDCS R3,R3,R2
        SUBS R5,R5,#1
        BNE ODD
STOP B STOP
        END
```

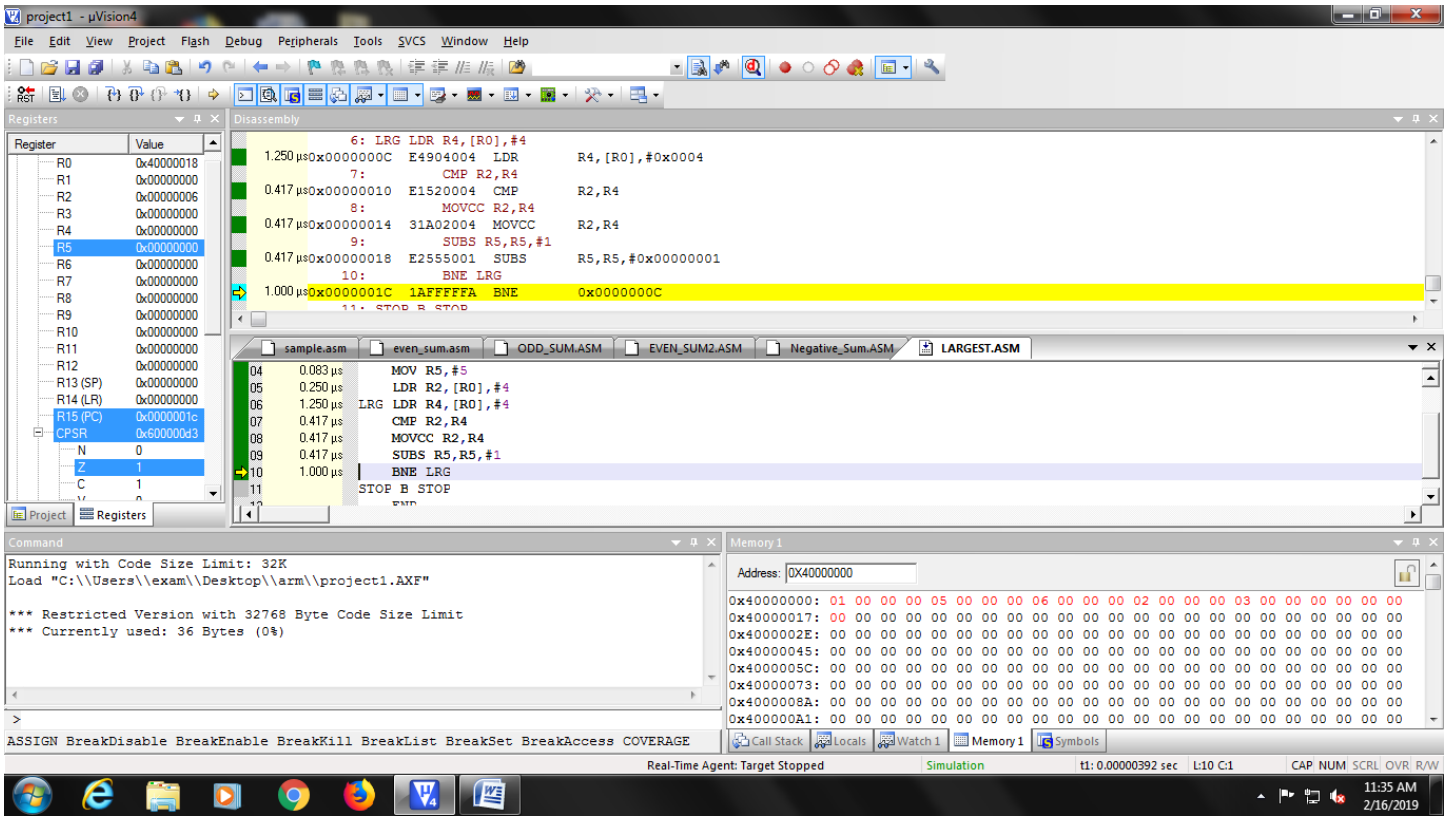


CODE 2:

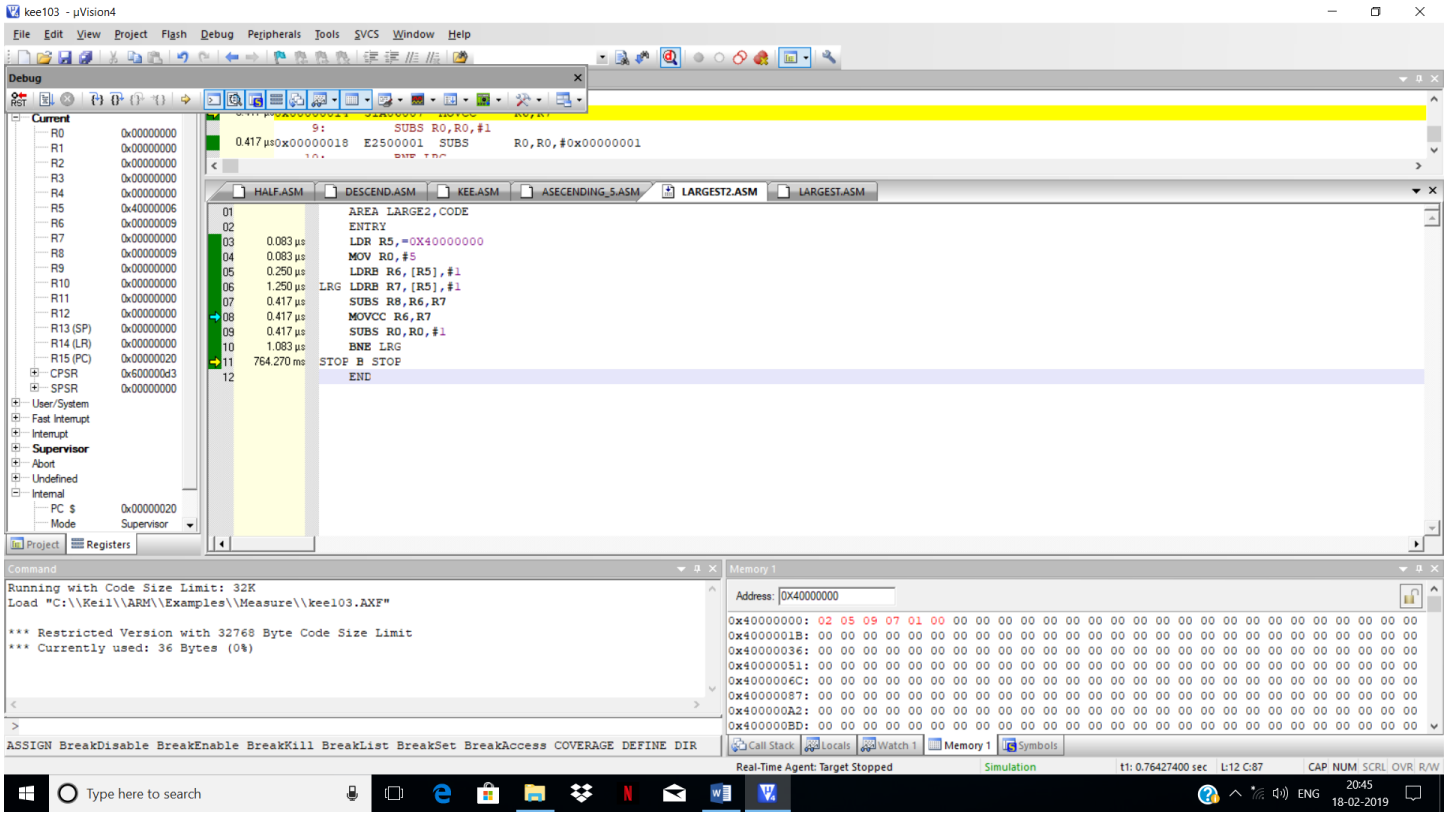
```
AREA SUMODD2,CODE
        ENTRY
        MOV R3,#0
        MOV R5,#05
        MOV R1,#00000001
        LDR R0,=0X40000000
ODD LDR R2,[R0],#4
        MOV R6,R2
        AND R8,R1,R6
        MOVS R8,R8,LSR #1
        ADDCS R3,R3,R2
        SUBS R5,R5,#1
        BNE ODD
STOP B STOP
        END
```



```
//ASM TO FIND THE LARGEST NUMBER IN THE GIVEN ARRAY
CODE 1:AREA LARGE,CODE
        ENTRY
        LDR R0,=0X40000000
        MOV R5,#5
        LDR R2,[R0],#4
LRG LDR R4,[R0],#4
        CMP R2,R4
        MOVCC R2,R4
        SUBS R5,R5,#1
        BNE LRG
STOP B STOP
        END
```

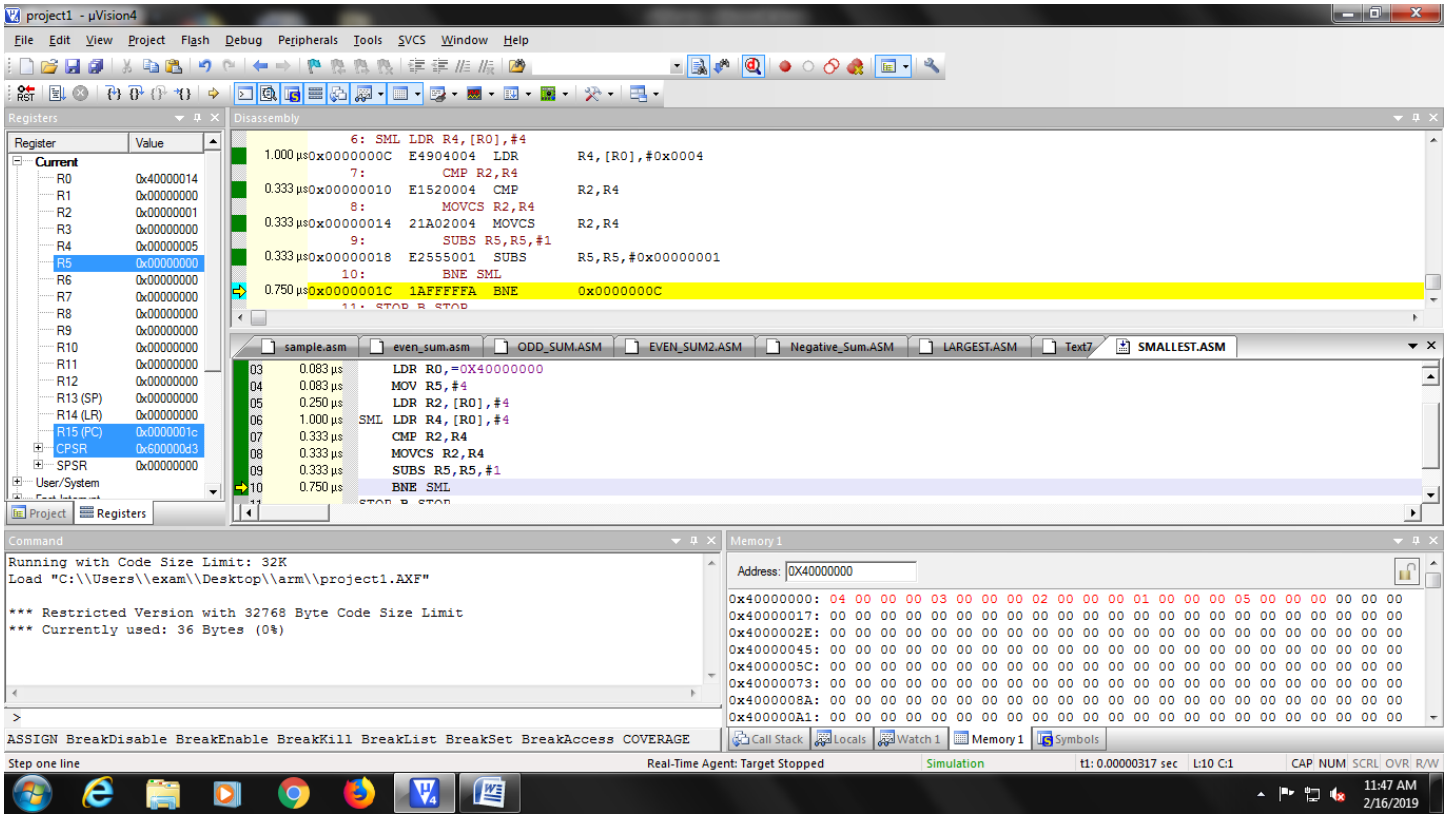


```
CODE 2:
        AREA LARGE2,CODE
        ENTRY
        LDR R5,=0X40000000
        MOV R0,#5
        LDRB R6,[R5],#1
LRG LDRB R7,[R5],#1
        SUBS R8,R6,R7
        MOVCC R6,R7
        SUBS R0,R0,#1
        BNE LRG
STOP B STOP
        END
```



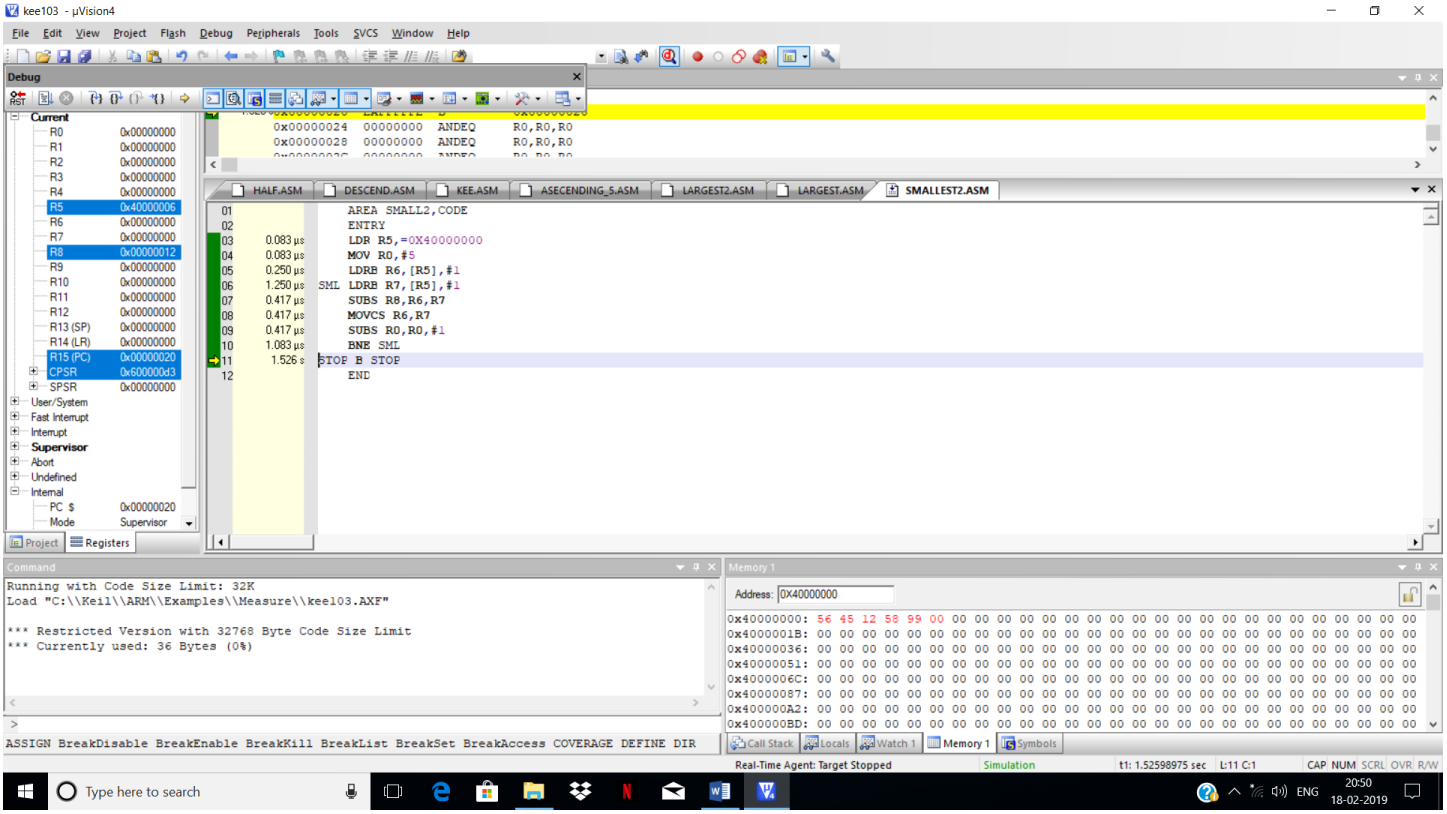

```
//ASM TO FIND THE SMALLEST NUMBER IN THE GIVEN ARRAY
CODE 1:AREA SMALL,CODE
        ENTRY
```

```
        LDR R0,=0X40000000
        MOV R5,#4
        LDR R2,[R0],#4
SML LDR R4,[R0],#4
        CMP R2,R4
        MOVCS R2,R4
        SUBS R5,R5,#1
        BNE SML
STOP B STOP
        END
```

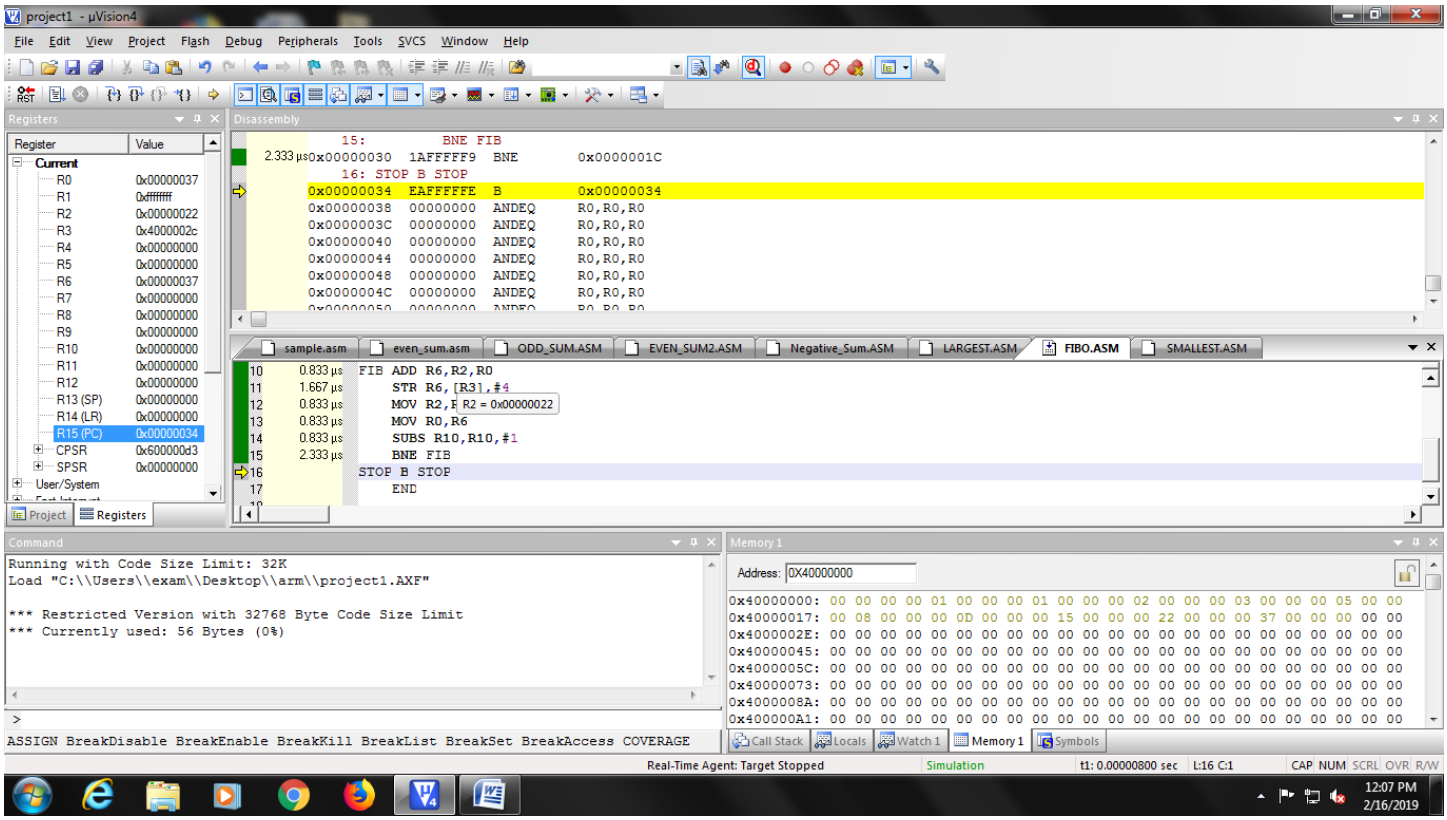


CODE 2:

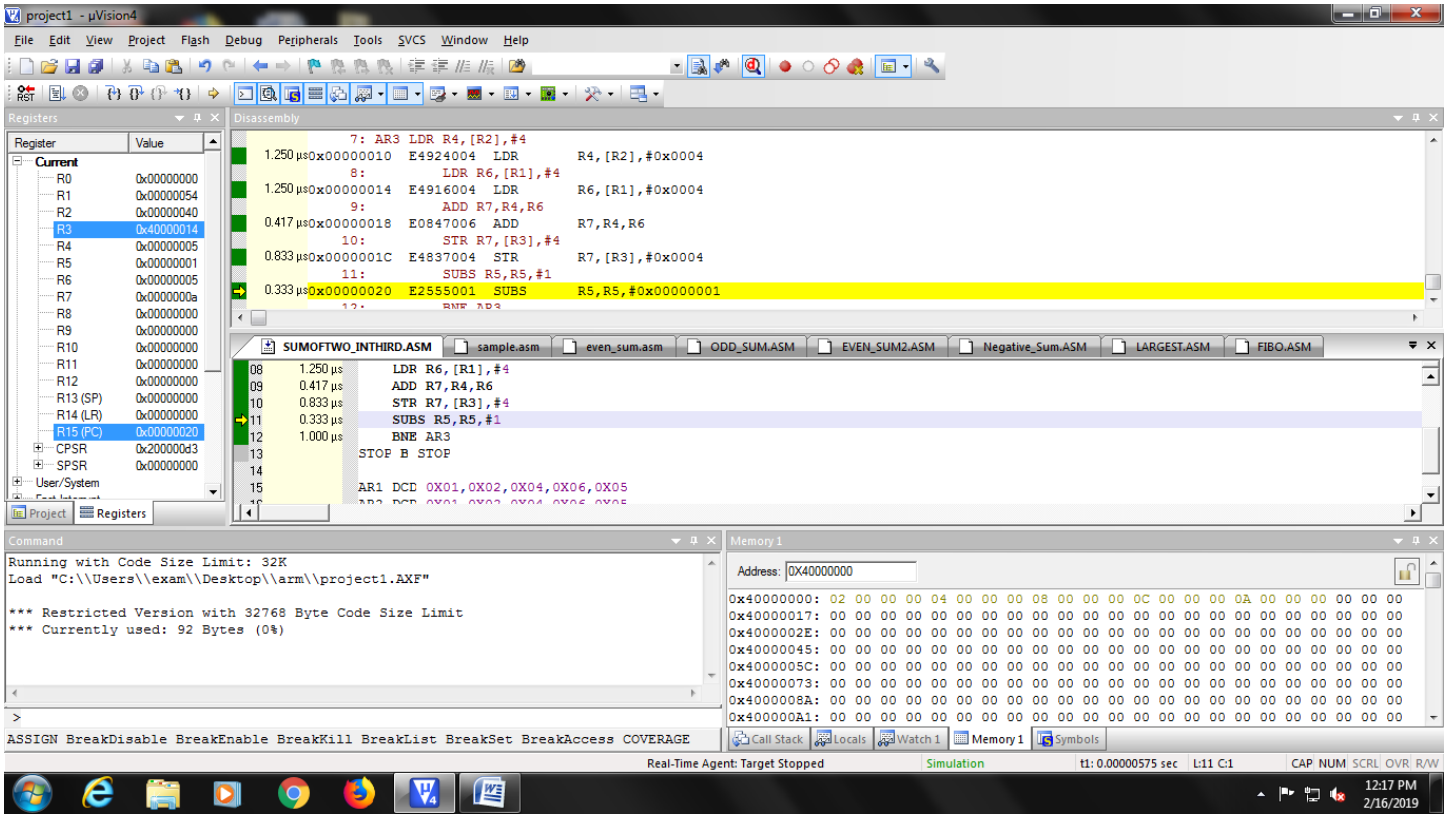
```
        AREA SMALL2,CODE
        ENTRY
        LDR R5,=0X40000000
        MOV R0,#5
        LDRB R6,[R5],#1
SML LDRB R7,[R5],#1
        SUBS R8,R6,R7
        MOVCS R6,R7
        SUBS R0,R0,#1
        BNE SML
STOP B STOP
        END
```


kee103 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Debug
Current
R0 0x00000000
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x40000006
R6 0x00000000
R7 0x00000000
R8 0x00000012
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000020
CPSR 0x600000d3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000020
Mode Supervisor
Project
Registers
0x00000024 00000000 ANDEQ R0,R0,R0
0x00000028 00000000 ANDEQ R0,R0,R0
HALF.ASM
DESCEND.ASM
KEE.ASM
ASECENDING_5.ASM
LARGEST2.ASM
LARGEST.ASM
SMALLEST2.ASM
01 AREA SMALL2,CODE
02 ENTRY
03 0.083 µs LDR R5,=0X40000000
04 0.083 µs MOV R0,#5
05 0.250 µs LDRB R6,[R5],#1
06 1.250 µs SML LDRB R7,[R5],#1
07 0.417 µs SUBS R8,R6,R7
08 0.417 µs MOVCS R6,R7
09 0.417 µs SUBS R0,R0,#1
10 1.083 µs BNE SML
11 1.526 s STOP B STOP
12 END
Command
Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\Examples\\Measure\\kee103.AXF"
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 36 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR
Memory 1
Address: 0X40000000
0x40000000: 56 45 12 58 99 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000001B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000051: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000006C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000087: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000A2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000BD: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack
Locals
Watch 1
Memory 1
Symbols
Real-Time Agent: Target Stopped
Simulation
t1: 1.52598975 sec
L:11 C:1
CAP NUM SCRL OVR R/W
Type here to search
ENG
20:50
18-02-2019

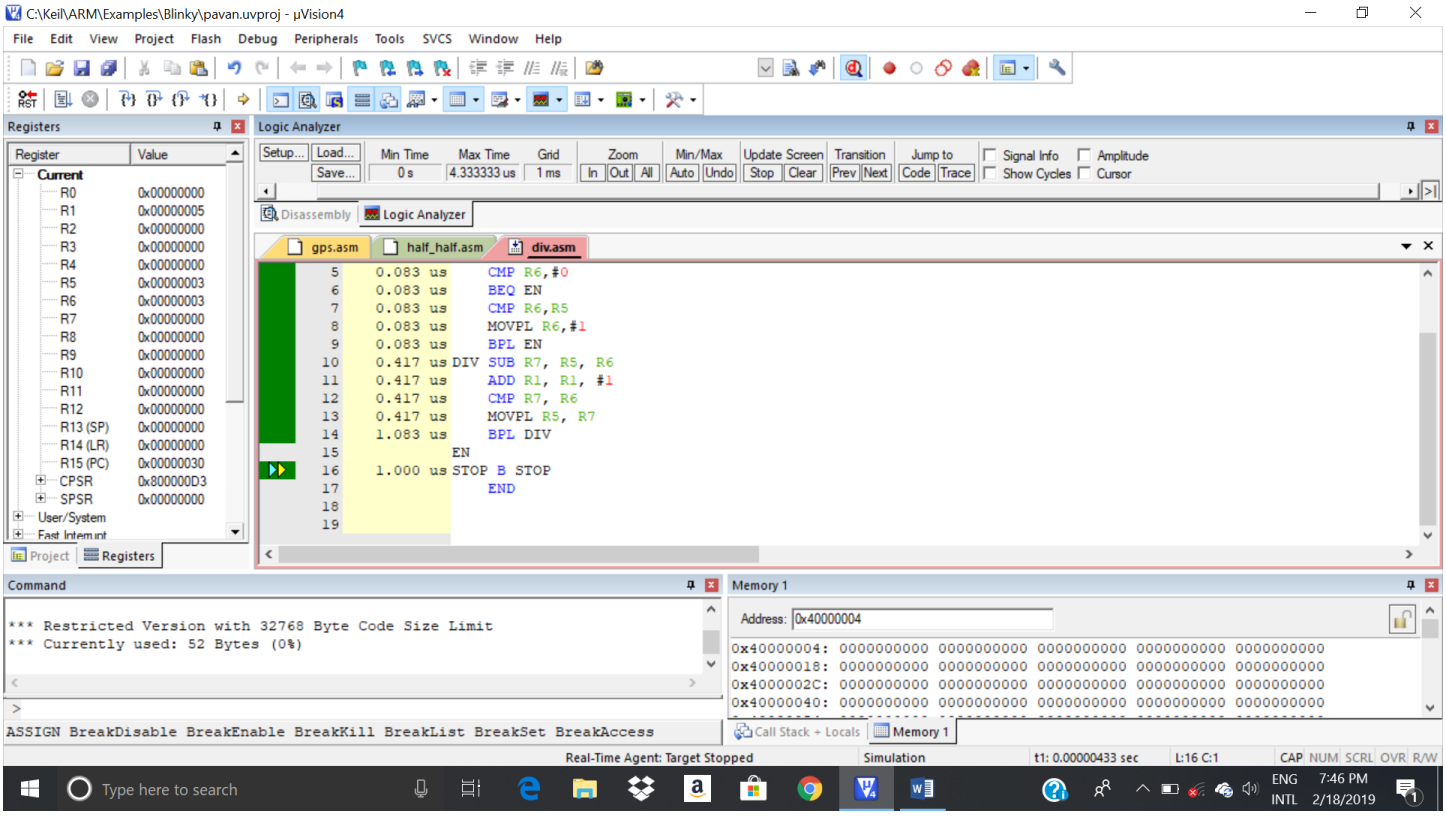
```
//FIBONACCI SERIES OF 10 NUMBERS
        AREA FIBONACCI,CODE
        ENTRY
        MOV R10,#10
        LDR R3,=0X40000000
        MOV R0,#0
        SUB R1,R0,#1
        MOV R2,#1
        ADD R6,R2,R1
        STR R6,[R3],#4
FIB     ADD R6,R2,R0
        STR R6,[R3],#4
        MOV R2,R0
        MOV R0,R6
        SUBS R10,R10,#1
        BNE FIB
STOP B STOP
        END
```



```
//TO ADD 2 ARRAY ELEMENTS AND STORE IN THIRD ARRAY
        AREA SUMOFTWO_INTHIRD,CODE
        ENTRY
        LDR R3,=0X40000000
        LDR R2,=AR1
        LDR R1,=AR2
        MOV R5,#5
AR3 LDR R4,[R2],#4
        LDR R6,[R1],#4
        ADD R7,R4,R6
        STR R7,[R3],#4
        SUBS R5,R5,#1
        BNE AR3
STOP B STOP
AR1 DCD 0X01,0X02,0X04,0X06,0X05
AR2 DCD 0X01,0X02,0X04,0X06,0X05
        END
```



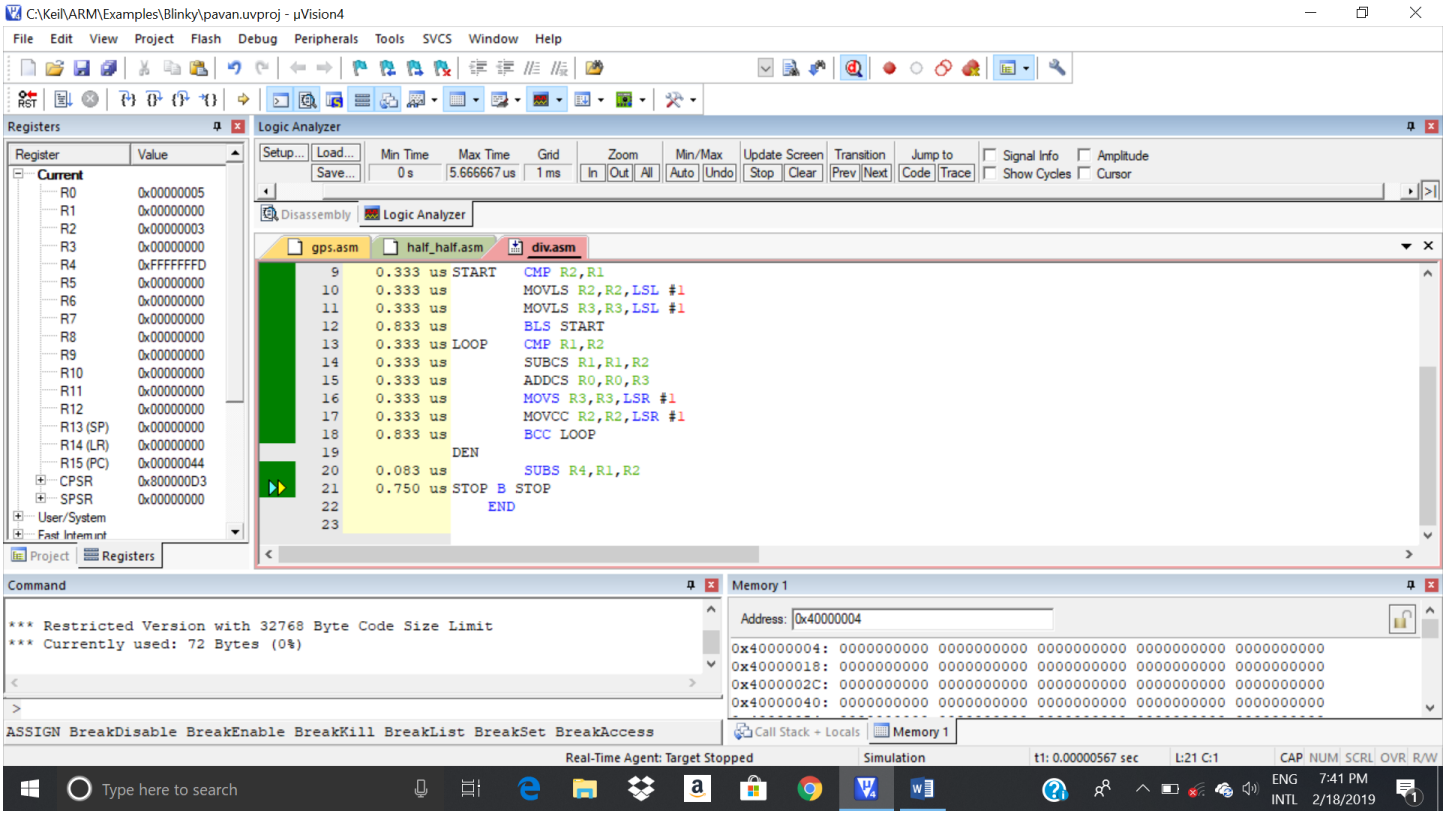
```
//DIVISION USING REPEATED SUBTRACTION
AREA A0, CODE
        ENTRY
        MOV R5, #15
        MOV R6, #3
        CMP R6, #0
        BEQ EN
        CMP R6, R5
        MOVPL R6, #1
        BPL EN
DIV     SUB R7, R5, R6
        ADD R1, R1, #1
        CMP R7, R6
        MOVPL R5, R7
        BPL DIV
EN
STOP B STOP
        END
```



CODE 2: DIVISION USING SHIFTING METHOD

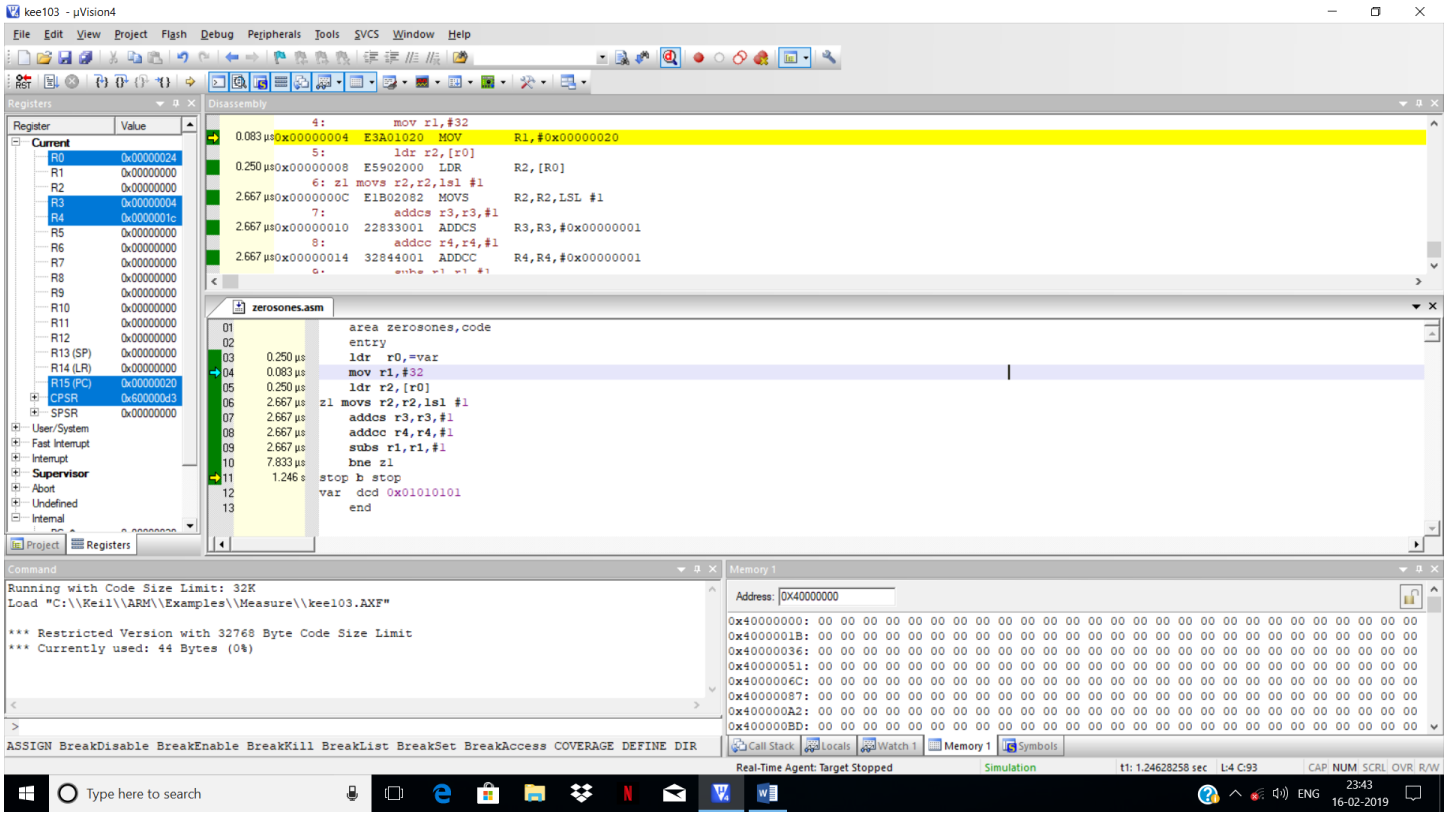
```
AREA A0, CODE
        ENTRY
        MOV R1, #15
        MOV R2, #3
        CMP R2, #0
        BEQ DEN
        MOV R0, #0
        MOV R3, #1
START   CMP R2, R1
        MOVLS R2, R2, LSL #1
```

```
                MOVLS R3, R3, LSL #1
                BLS START
LOOP            CMP R1, R2
                SUBCS R1, R1, R2
                ADDCS R0, R0, R3
                MOVS R3, R3, LSR #1
                MOVCC R2, R2, LSR #1
                BCC LOOP
DEN
                SUBS R4, R1, R2
STOP B STOP
        END
```

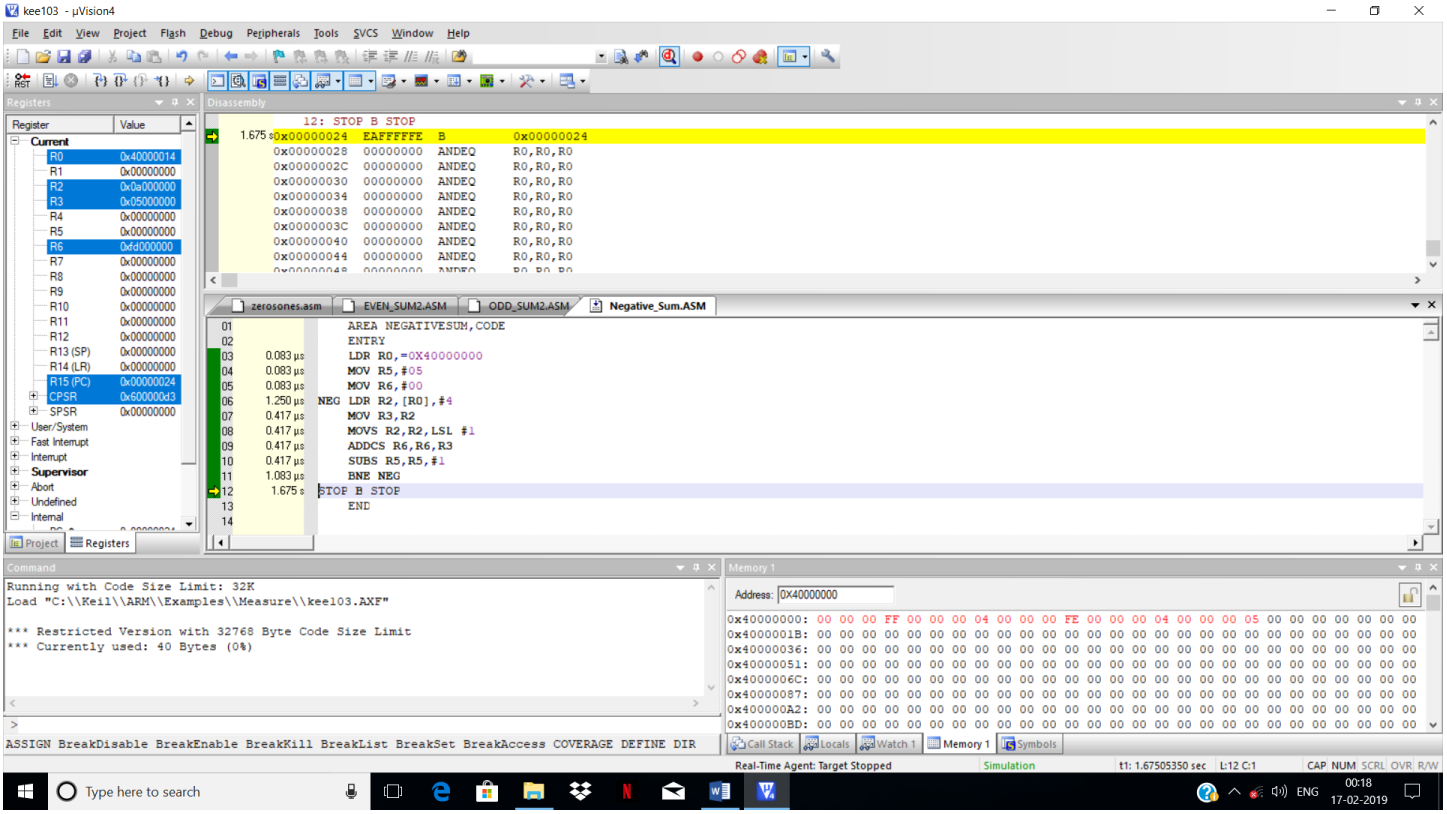



```
// ASSEMBLY PROGRAM TO COUNT NO OF ZEROS AND ONES
        AREA ZEROS_ONES,CODE
        ENTRY
        LDR  R0,=VAR
        MOV R1,#32
        LDR R2,[R0]
Z1 MOVS R2,R2,LSL #1
        ADDCS R3,R3,#1
        ADDCC R4,R4,#1
```

```
        SUBS R1,R1,#1
        BNE Z1
STOP B STOP
VAR  DCD 0X01010101
        END
```

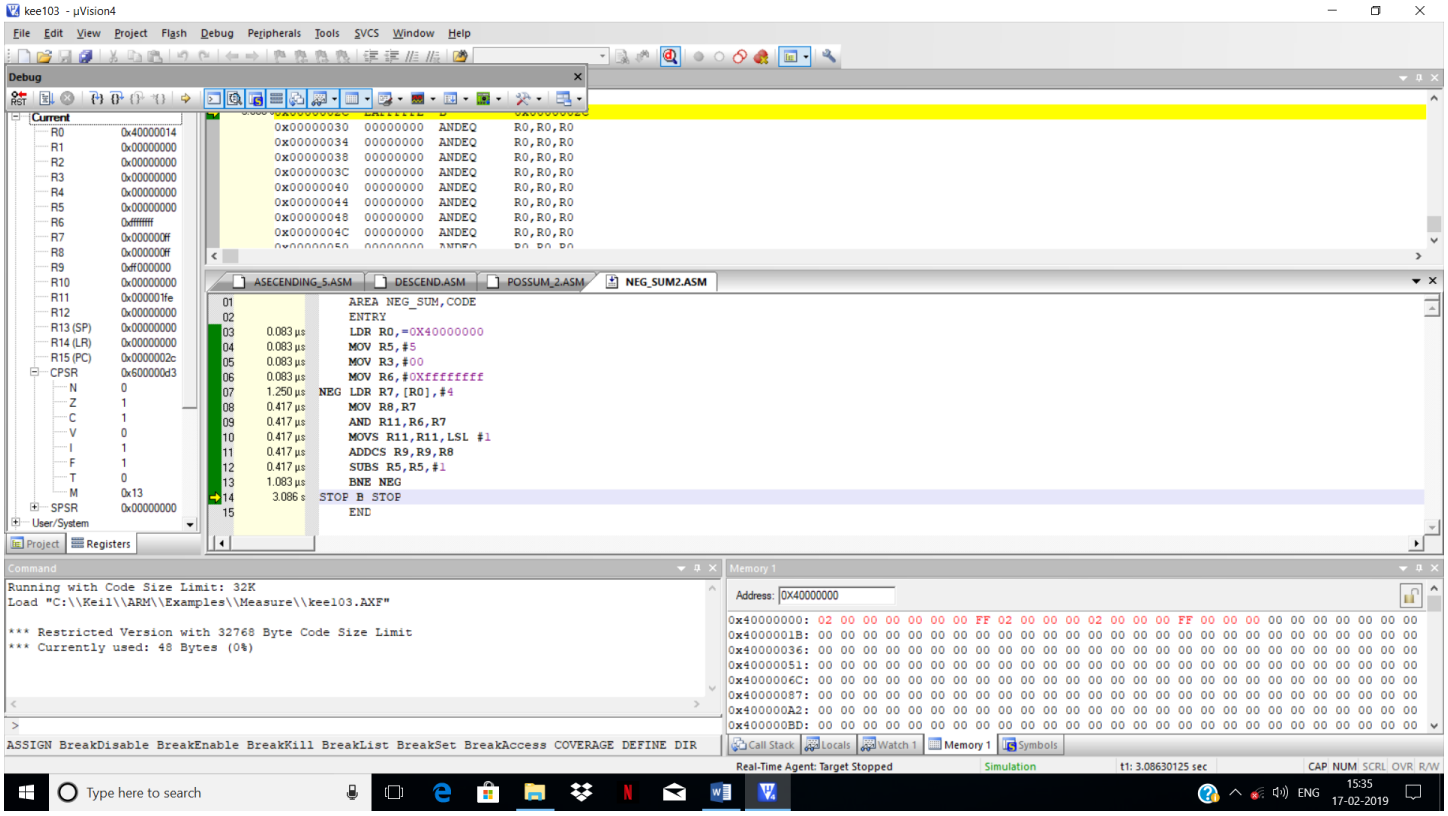



```
// PROGRAM TO FIND SUM OF NEGATIVE ELEMENTS IN ARRAY
CODE 1:
        AREA NEGATIVESUM,CODE
        ENTRY
        LDR R0,=0X40000000
        MOV R5,#05
        MOV R6,#00
NEG LDR R2,[R0],#4
        MOV R3,R2
        MOVS R2,R2,LSL #1
        ADDCS R6,R6,R3
        SUBS R5,R5,#1
        BNE NEG
STOP B STOP
        END
```


kee103 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x40000014
R1 0x00000000
R2 0x0a000000
R3 0x05000000
R4 0x00000000
R5 0x00000000
R6 0xfd000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000024
CPSR 0x600000d3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
Project Registers
Disassembly
12: STOP B STOP
1.675 s 0x00000024 EAFFFFFE B 0x00000024
0x00000028 00000000 ANDEQ R0,R0,R0
0x0000002C 00000000 ANDEQ R0,R0,R0
0x00000030 00000000 ANDEQ R0,R0,R0
0x00000034 00000000 ANDEQ R0,R0,R0
0x00000038 00000000 ANDEQ R0,R0,R0
0x0000003C 00000000 ANDEQ R0,R0,R0
0x00000040 00000000 ANDEQ R0,R0,R0
0x00000044 00000000 ANDEQ R0,R0,R0
zerosones.asm EVEN_SUM2.ASM ODD_SUM2.ASM Negative_Sum.ASM
01 AREA NEGATIVESUM,CODE
02 ENTRY
03 0.083 µs LDR R0,=0X40000000
04 0.083 µs MOV R5,#05
05 0.083 µs MOV R6,#00
06 1.250 µs NEG LDR R2,[R0],#4
07 0.417 µs MOV R3,R2
08 0.417 µs MOVS R2,R2,LSL #1
09 0.417 µs ADDCS R6,R6,R3
10 0.417 µs SUBS R5,R5,#1
11 1.083 µs BNE NEG
12 1.675 s STOP B STOP
13 END
14
Command
Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\Examples\\Measure\\kee103.AXF"
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 40 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR
Memory 1
Address: 0X40000000
0x40000000: 00 00 00 FF 00 00 00 04 00 00 00 FE 00 00 00 04 00 00 00 05 00 00 00 00 00 00 00
0x4000001B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000051: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000006C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000087: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000A2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000BD: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack Locals Watch 1 Memory 1 Symbols
Real-Time Agent: Target Stopped
Simulation
t1: 1.67505350 sec
L:12 C:1
CAP NUM SCRL OVR R/W
Type here to search
ENG
00:18
17-02-2019

CODE 2:

```
        AREA NEG_SUM,CODE
        ENTRY
        LDR R0,=0X40000000
        MOV R5,#5
        MOV R3,#00
        MOV R6,#0Xffffffff
NEG LDR R7,[R0],#4
        MOV R8,R7
        AND R11,R6,R7
        MOVS R11,R11,LSL #1
        ADDCS R9,R9,R8
        SUBS R5,R5,#1
        BNE NEG
STOP B STOP
   END
```

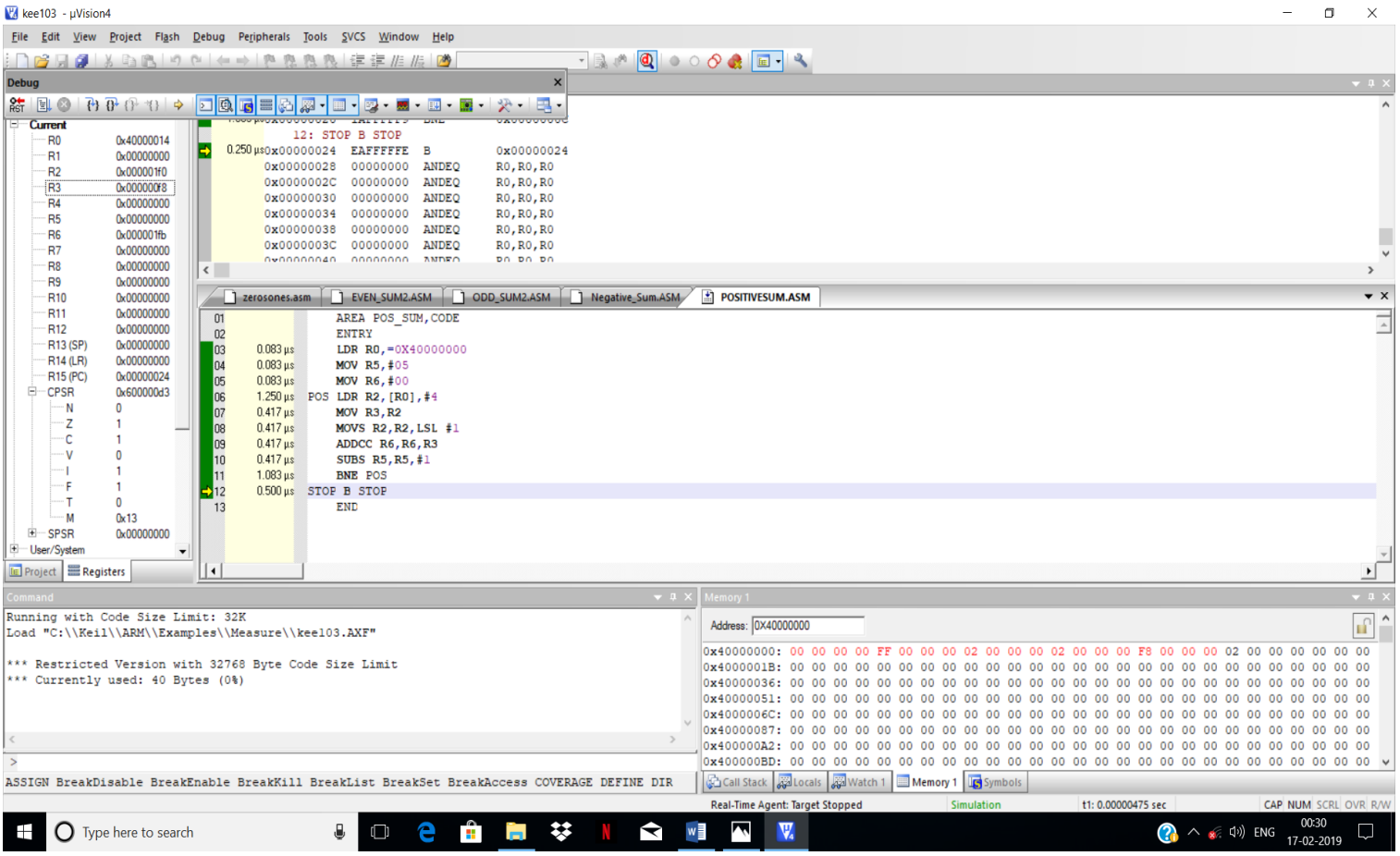


//PROGRAM TO FIND SUM OF POSITIVE NUMBERS IN ARRAY

CODE 1:

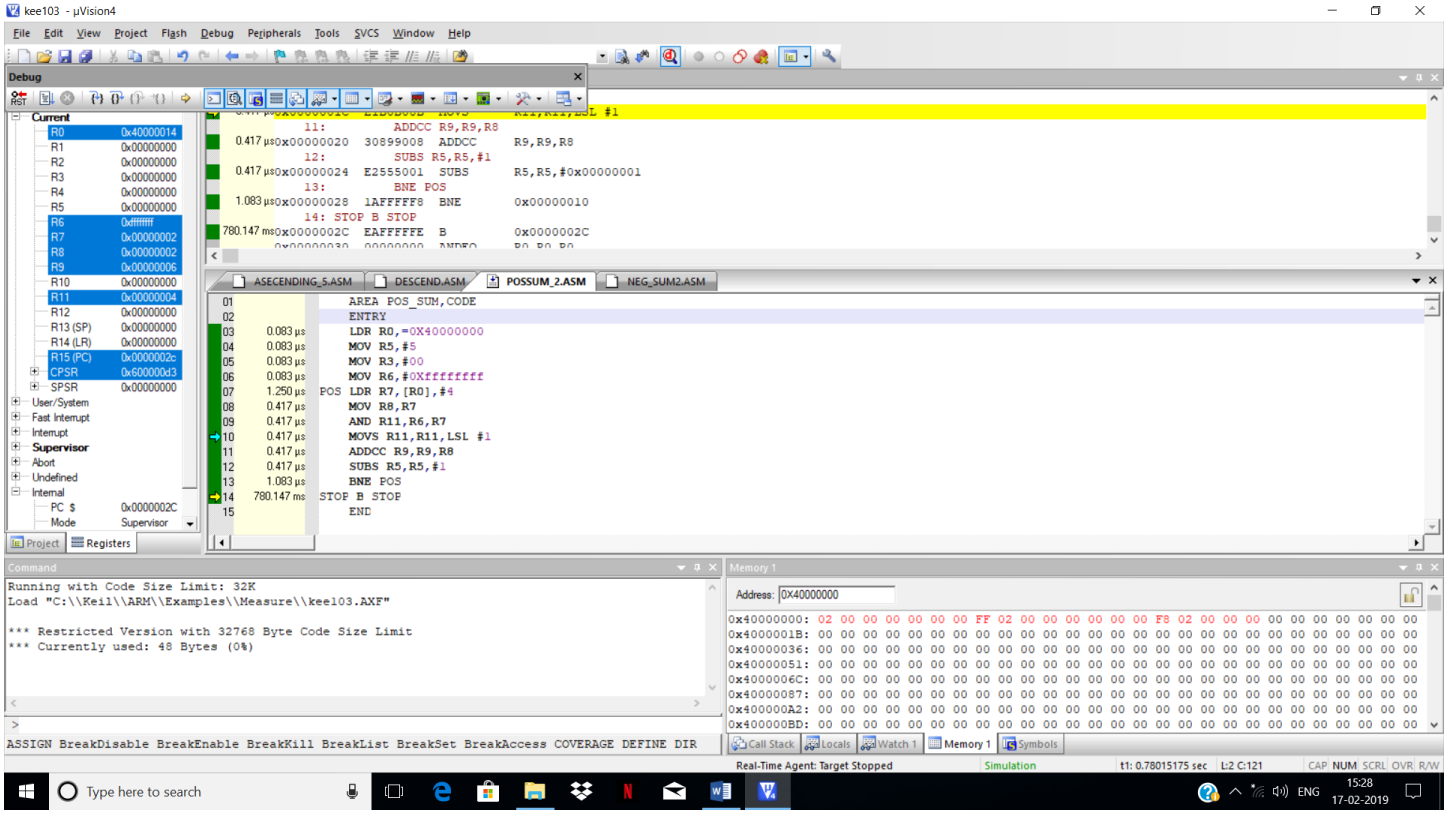
```
AREA POS_SUM,CODE
        ENTRY
        LDR R0,=0X40000000
        MOV R5,#05
        MOV R6,#00
POS LDR R2,[R0],#4
        MOV R3,R2
        MOVS R2,R2,LSL #1
        ADDCC R6,R6,R3
        SUBS R5,R5,#1
```

```
        BNE POS
STOP B STOP
        END
```



CODE 2:

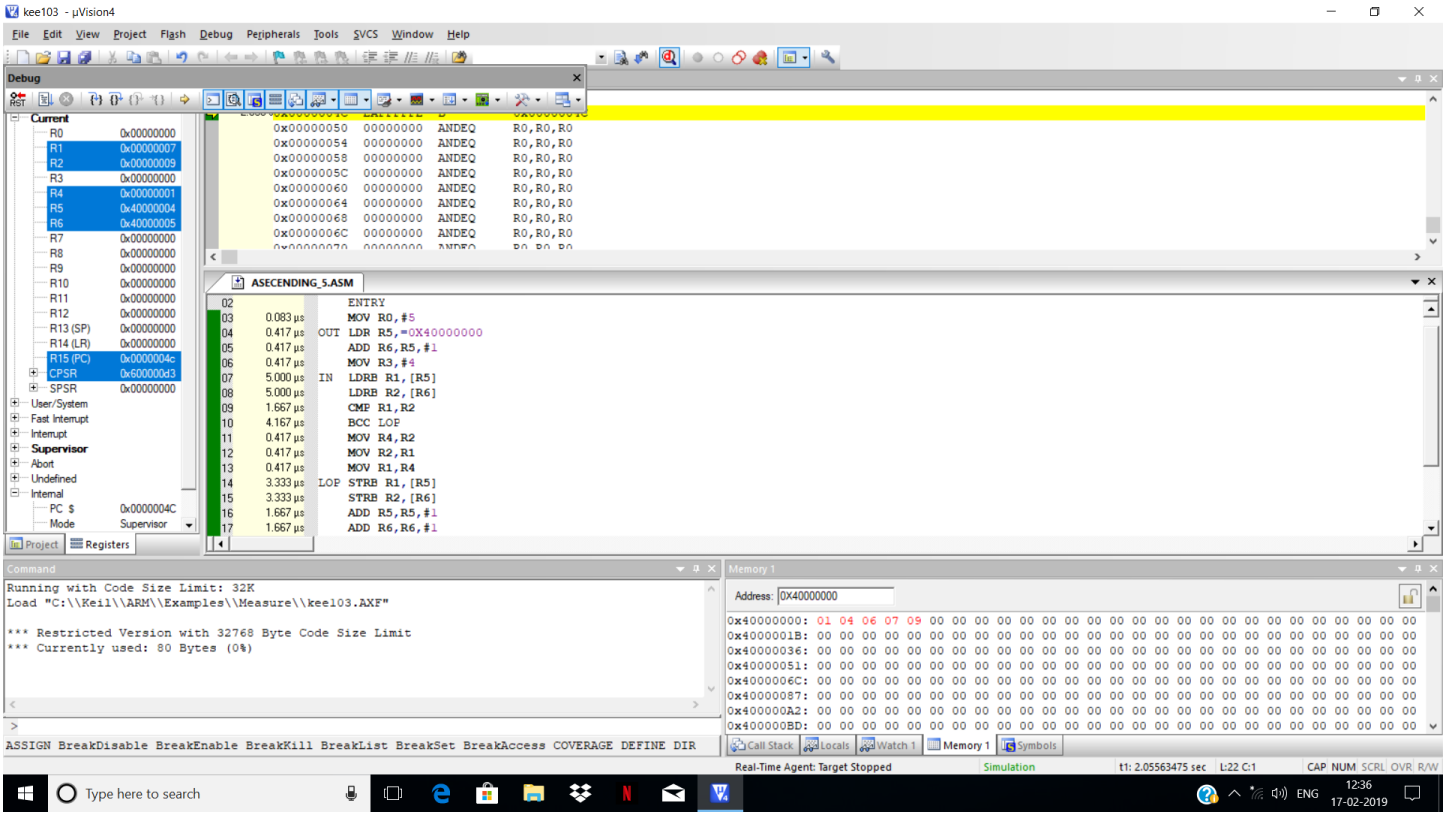
```
        AREA POS_SUM,CODE
        ENTRY
        LDR R0,=0X40000000
        MOV R5,#5
        MOV R3,#00
        MOV R6,#0Xffffffff
POS LDR R7,[R0],#4
        MOV R8,R7
        AND R11,R6,R7
        MOVS R11,R11,LSL #1
        ADDCC R9,R9,R8
        SUBS R5,R5,#1
        BNE POS
STOP B STOP
   END
```


kee103 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Debug
Current
R0 0x40000014
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x00000000
R6 0xffffffff
R7 0x00000002
R8 0x00000002
R9 0x00000006
R10 0x00000000
R11 0x00000004
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x0000002c
CPSR 0x600000d3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000002C
Mode Supervisor
Project
Registers
11: ADDCC R9,R9,R8
0.417 µs 0x00000020 30899008 ADDCC R9,R9,R8
12: SUBS R5,R5,#1
0.417 µs 0x00000024 E2555001 SUBS R5,R5,#0x00000001
13: BNE POS
1.083 µs 0x00000028 1AFFFFF8 BNE 0x00000010
14: STOP B STOP
780.147 ms 0x0000002C EAFFFFFE B 0x0000002C
ASCENDING_5.ASM
DESCEND.ASM
POSSUM_2.ASM
NEG_SUM2.ASM
01 AREA POS_SUM,CODE
02 ENTRY
03 0.083 µs LDR R0,=0X40000000
04 0.083 µs MOV R5,#5
05 0.083 µs MOV R3,#00
06 0.083 µs MOV R6,#0Xffffffff
07 1.250 µs POS LDR R7,[R0],#4
08 0.417 µs MOV R8,R7
09 0.417 µs AND R11,R6,R7
10 0.417 µs MOVS R11,R11,LSL #1
11 0.417 µs ADDCC R9,R9,R8
12 0.417 µs SUBS R5,R5,#1
13 1.083 µs BNE POS
14 780.147 ms STOP B STOP
15 END
Command
Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\Examples\\Measure\\kee103.AXF"
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 48 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR
Memory 1
Address: 0X40000000
0x40000000: 02 00 00 00 00 00 00 FF 02 00 00 00 00 00 00 F8 02 00 00 00 00 00 00 00 00 00
0x4000001B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000051: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000006C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000087: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000A2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000BD: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack
Locals
Watch 1
Memory 1
Symbols
Real-Time Agent: Target Stopped
Simulation
t1: 0.78015175 sec
L:2 C:121
CAP NUM SCRL OVR R/W
Type here to search
ENG
15:28
17-02-2019

//ASSEMBLY PROGRAM TO ARRANGE THE NUMBERS IN ASCENDING ORDER

CODE 1:

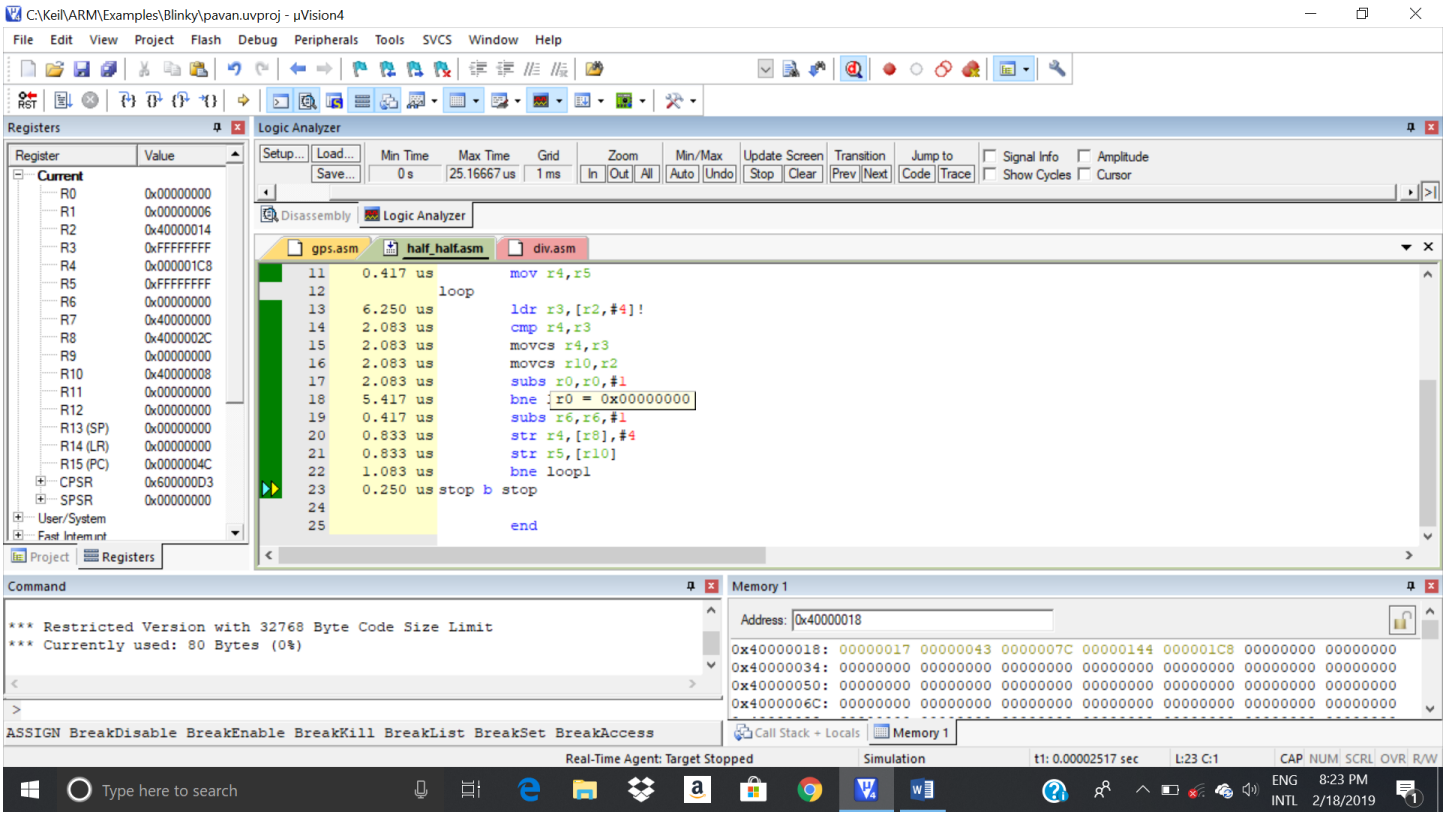
```
AREA ASECEND,CODE
        ENTRY
        MOV R0,#5
OUT LDR R5,=0X40000000
        ADD R6,R5,#1
        MOV R3,#4
IN  LDRB R1,[R5]
        LDRB R2,[R6]
        CMP R1,R2
        BCC LOP
        MOV R4,R2
        MOV R2,R1
        MOV R1,R4
LOP     STRB R1,[R5]
        STRB R2,[R6]
        ADD R5,R5,#1
        ADD R6,R6,#1
        SUBS R3,R3,#1
        BNE IN
        SUBS R0,R0,#1
        BNE OUT
STOP B STOP
        END
```



CODE 2:

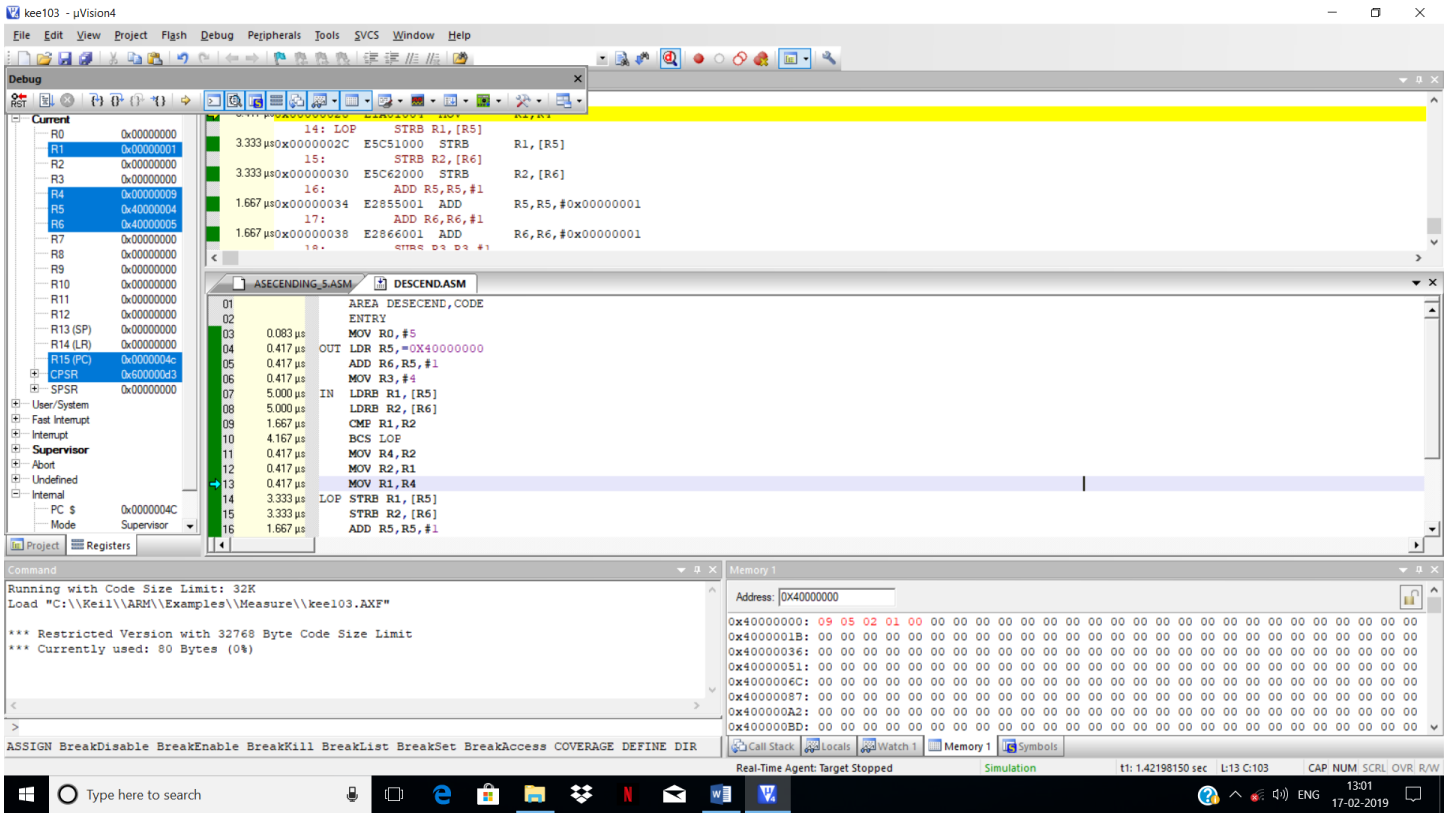
```
        AREA A0,CODE
        ENTRY
```

```
                MOV R6,#5
                MOV R0,#0
                MOV R1,#6
                MOV R5,#0XFFFFFFFF
                LDR R7,=0X40000000
                LDR R8,=0X40000018
LOOP1           SUB R0,R1,#1
                MOV R2,R7
                MOV R4,R5
LOOP
                LDR R3,[R2,#4]!
                CMP R4,R3
                MOVCS R4,R3
                MOVCS R10,R2
                SUBS R0,R0,#1
                BNE LOOP
                SUBS R6,R6,#1
                STR R4,[R8],#4
                STR R5,[R10]
                BNE LOOP1
STOP B STOP
                END
```



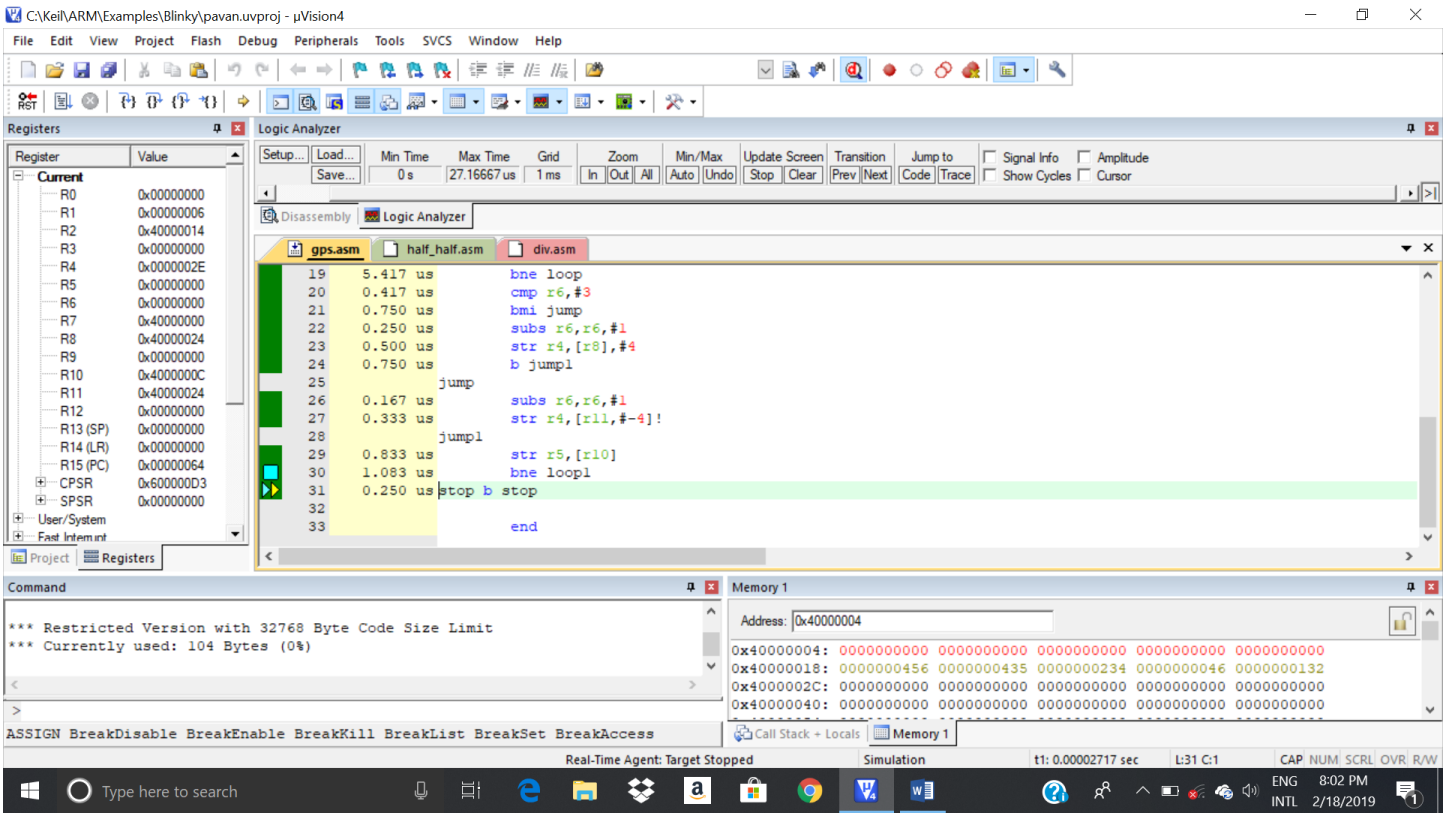

```
// ASSEMBLY PROGRAM TO ARRANGE NUMBERS IN DESCENDING
CODE 1:         AREA DESCEND,CODE
        ENTRY
        MOV R0,#5
OUT LDR R5,=0X40000000
        ADD R6,R5,#1
        MOV R3,#4
```

```
IN  LDRB R1,[R5]
        LDRB R2,[R6]
        CMP R1,R2
        BCS LOP
        MOV R4,R2
        MOV R2,R1
        MOV R1,R4
LOP     STRB R1,[R5]
        STRB R2,[R6]
        ADD R5,R5,#1
        ADD R6,R6,#1
        SUBS R3,R3,#1
        BNE IN
        SUBS R0,R0,#1
        BNE OUT
STOP B STOP
        END
```




```
CODE 2: AREA A0,CODE
                ENTRY
                MOV R6,#5
                MOV R0,#0
                MOV R1,#6
                MOV R5,#0
                LDR R7,=0X40000000
                LDR R8,=0X40000018
LOOP1           SUB R0,R1,#1
                MOV R2,R7
```

```
                MOV R4,R5
LOOP
                LDR R3,[R2,#4]!
                CMP R4,R3
                MOVCC R4,R3
                MOVCC R10,R2
                SUBS R0,R0,#1
                BNE LOOP
                SUBS R6,R6,#1
                STR R4,[R8],#4
                STR R5,[R10]
                BNE LOOP1
STOP B STOP
END
```

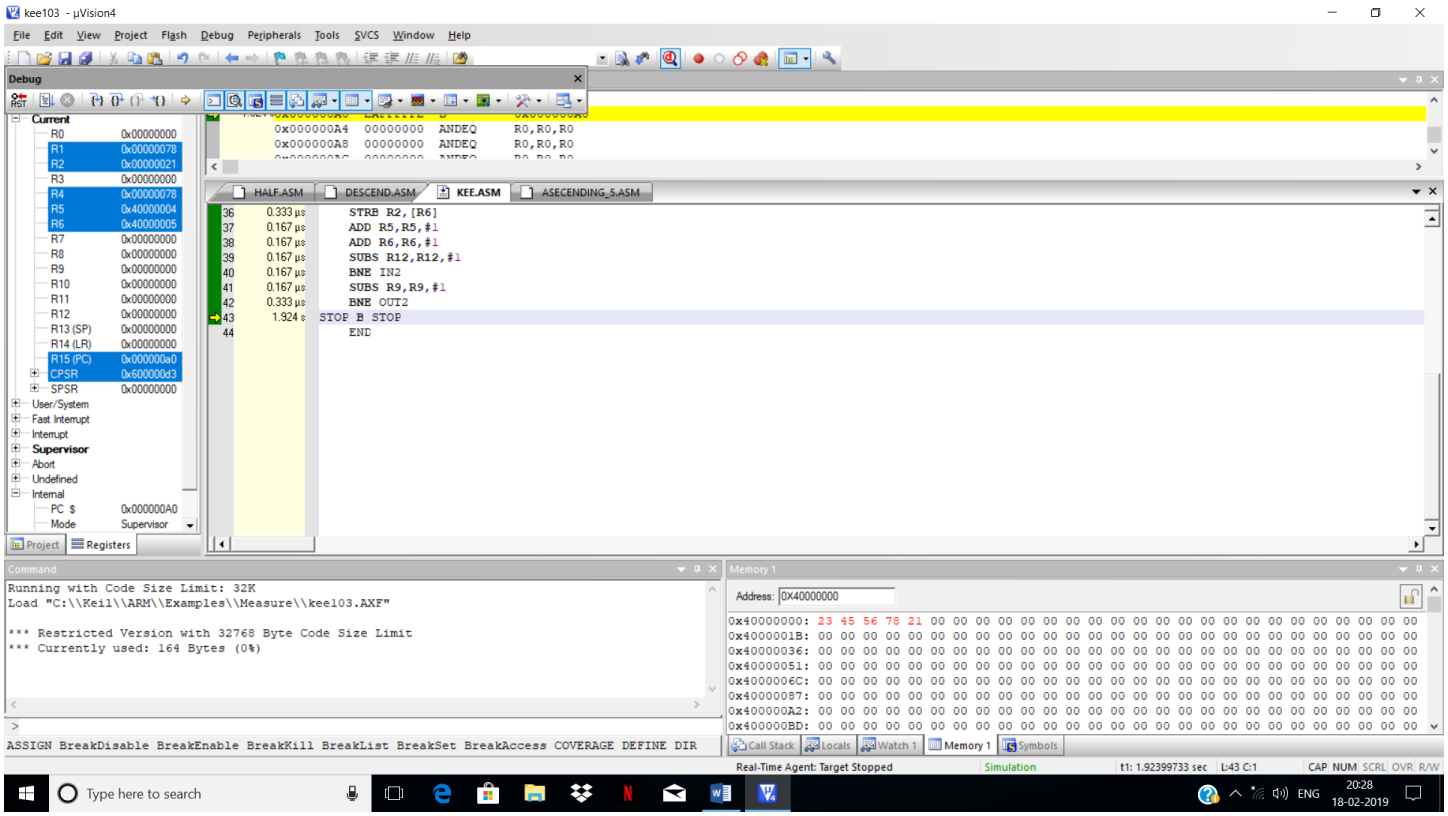


// ASSEMBLY PROGRAM TO ARRANGE THE NUMBERS IN HALF ASCENDING AND HALF DESCENDING ORDER

CODE 1:

```
   AREA HALF,CODE
        ENTRY
        MOV R0,#3
        MOV R9,#2
        MOV R10,#0
OUT1 LDR R5,=0X40000000
```

```
        ADD R6,R5,#1
        MOV R3,#2
IN1  LDRB R1,[R5]
        LDRB R2,[R6]
        CMP R1,R2
        BCC LOP
        MOV R4,R2
        MOV R2,R1
        MOV R1,R4
LOP     STRB R1,[R5]
        STRB R2,[R6]
        ADD R5,R5,#1
        ADD R6,R6,#1
        SUBS R3,R3,#1
        BNE IN1
        SUBS R0,R0,#1
   BNE OUT1
        BEQ OUT2
OUT2 LDR R5,=0X40000003
         ADD R6,R5,#1
         MOV R12,#1
IN2 LDRB R1,[R5]
        LDRB R2,[R6]
        CMP R1,R2
        BCS LOP2
        MOV R4,R2
        MOV R2,R1
        MOV R1,R4
LOP2 STRB R1,[R5]
        STRB R2,[R6]
        ADD R5,R5,#1
        ADD R6,R6,#1
        SUBS R12,R12,#1
        BNE IN2
        SUBS R9,R9,#1
        BNE OUT2
STOP B STOP
        END
```




```
CODE 2: AREA A0,CODE
                ENTRY
                MOV R6,#5
                MOV R0,#0
                MOV R1,#6
                MOV R5,#0
                LDR R7,=0X40000000
                LDR R8,=0X40000018
                LDR R11,=0X4000002C
LOOP1           SUB R0,R1,#1
                MOV R2,R7
                MOV R4,R5
LOOP
                LDR R3,[R2,#4]!
                CMP R4,R3
                MOVCC R4,R3
                MOVCC R10,R2
                SUBS R0,R0,#1
                BNE LOOP
                CMP R6,#3
                BMI JUMP
                SUBS R6,R6,#1
                STR R4,[R8],#4
                B JUMP1
JUMP
                SUBS R6,R6,#1
                STR R4,[R11,#-4]!
JUMP1
```

```
        STR R5,[R10]
        BNE LOOP1
STOP B STOP
END
```