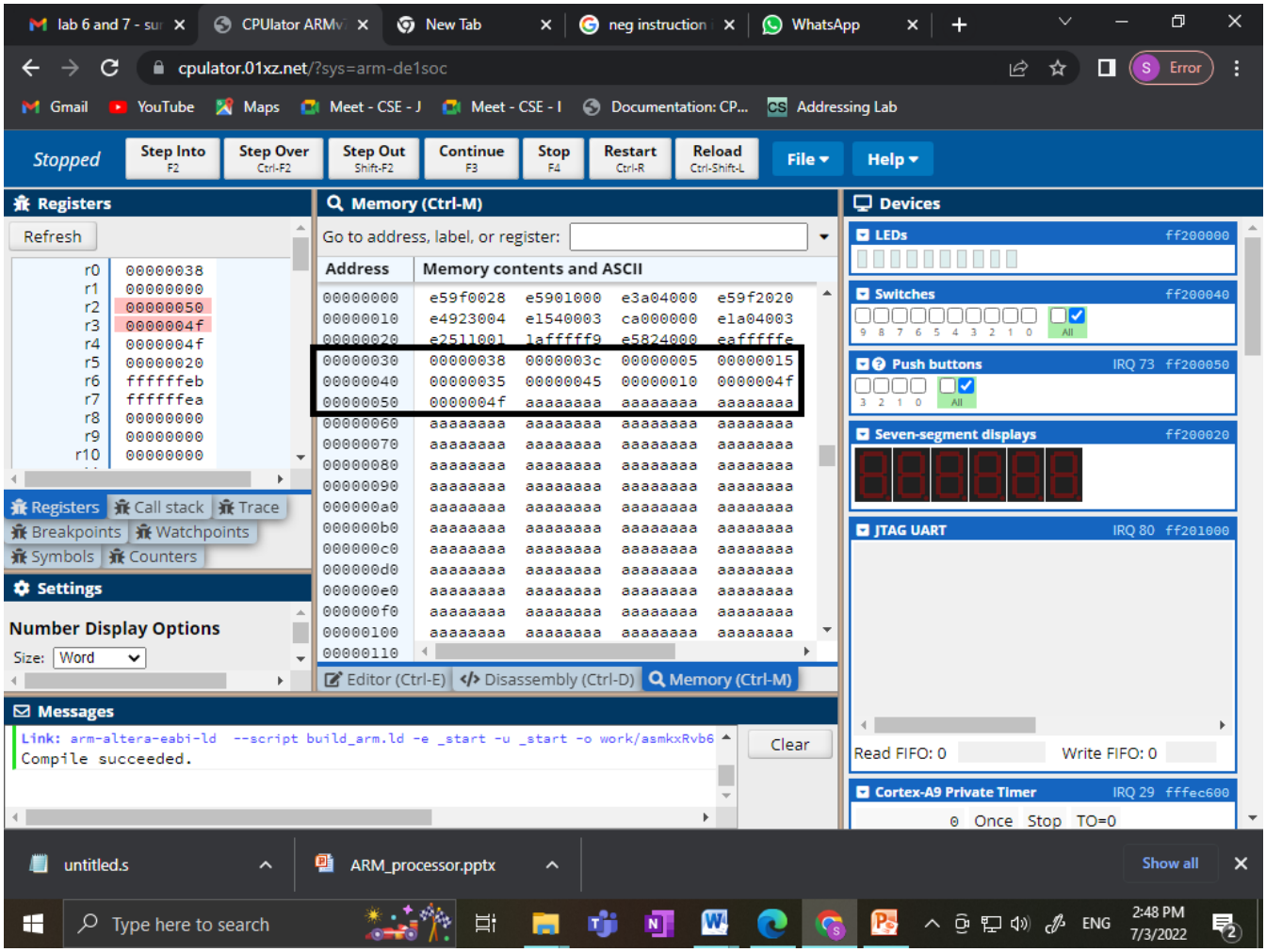
Lab-7:

Objective 1: Find the Largest number from a given array of size N using ARM assembly language.

Program:

```
.global _start
_start:
        @Largest number from a given array
        ldr r0,=count
        ldr r1,[r0]
        mov r4,#0x00
        ldr r2,=array
back: ldr r3, [r2],#4
        cmp r4,r3
         bgt fwd
        mov r4,r3
fwd:   subs r1,r1,#01
        bne back
        str r4,[r2]
exit: b exit

.data
 count: .word 0x05
 array: .word 0x15, 0x35, 0x45, 0x10,0x4f
Result:
```



Objective 1: Find the smallest number from an array of size N using ARM assembly language

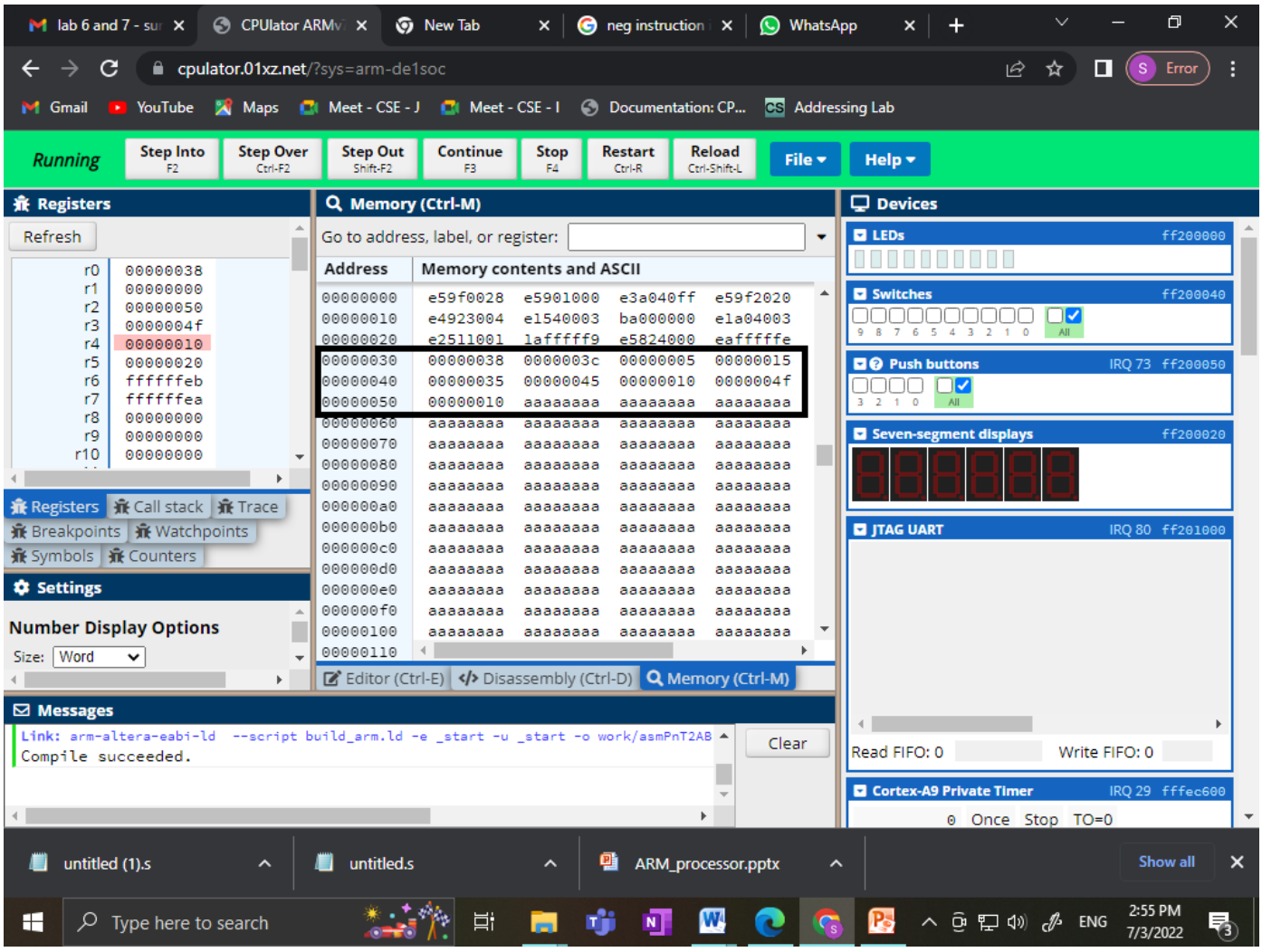
Program:

```
.global _start
_start:
        @Smallest number from a given array
        ldr r0,=count
        ldr r1,[r0]
        mov r4,#0x0ff
        ldr r2,=array
back: ldr r3, [r2],#4
        cmp r4,r3
         blt fwd
        mov r4,r3
```

```
fwd:   subs r1,r1,#01
       bne back
       str r4,[r2]
exit: b exit

.data
 count: .word 0x05
 array: .word 0x15, 0x35, 0x45, 0x10,0x4f
```

Result:



Objective-2 : Separate Even numbers and Odds numbers in a given array of size N using ARM microprocessor.

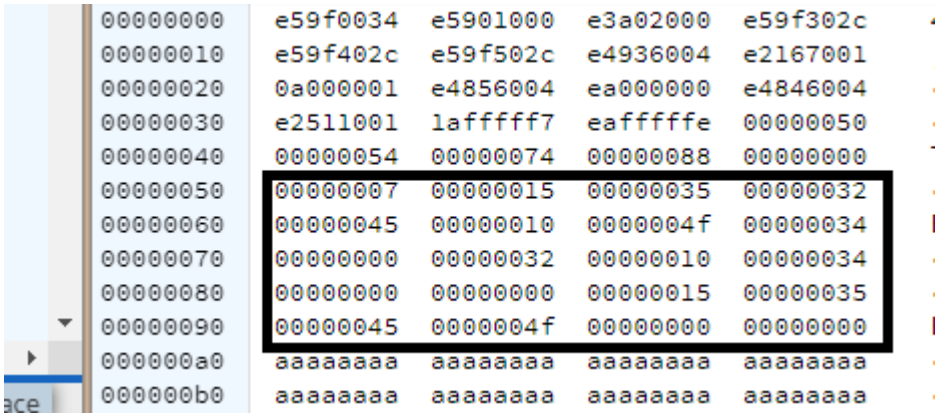
Program:

```
.global _start
_start:
        ldr r0,=count
        ldr r1,[r0]
        ldr r3,=array @ r3 = base address of array=array[0]
        ldr r4,=even  @ r4=base address of even data locations =even[0]
        ldr r5,=odd   @ r5=base address of odd data locations =odd[0]
back: ldr r6, [r3],#4
        ands r7,r6,#1
        beq fwd
        str r6,[r5],#4
         b fwd1
fwd:  str r6,[r4],#4
fwd1: subs r1,r1,#01
          bne back

exit: b exit

.data
 count: .word 0x07
 array: .word 0x15, 0x35,0x32, 0x45, 0x10,0x4f,0x34,
 even: .word 0, 0, 0, 0, 0
 odd: .word 0, 0, 0, 0, 0
```

Result:



00000000	e59f0034	e5901000	e3a02000	e59f302c
00000010	e59f402c	e59f502c	e4936004	e2167001
00000020	0a000001	e4856004	ea000000	e4846004
00000030	e2511001	1affffff7	eafffffe	00000050
00000040	00000054	00000074	00000088	00000000
00000050	00000007	00000015	00000035	00000032
00000060	00000045	00000010	0000004f	00000034
00000070	00000000	00000032	00000010	00000034
00000080	00000000	00000000	00000015	00000035
00000090	00000045	0000004f	00000000	00000000
000000a0	aaaaaaaa	aaaaaaaa	aaaaaaaa	aaaaaaaa
000000b0	aaaaaaaa	aaaaaaaa	aaaaaaaa	aaaaaaaa