

# ECE 322 EMBEDDED SYSTEM

## Lab :-5

Name :- ARVIND KUMAR

Roll N:-2021BEC0035

DATE : 06/02/2024

Submitted to - Della Ma'am

---

### Aim:-

5(a) – Sum of the element of given array if the element value is greater than 5.

### Code:

**Registers**

Register	Value
<b>Core</b>	
R0	0x0000003C
R1	0x00000000
R2	0x0000002C
R3	0x0000001E
R4	0x10000000
R5	0x00000000
R6	0x0000003C
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R...	0x20001000
R...	0xFFFFFFFF
R...	0x00000052
xP...	0x61000000
<b>Banked</b>	
<b>System</b>	
<b>Internal</b>	
M...	Thread
Pri...	Privileged
St...	MSP
St...	3217813
Sec	0.26815108

**lab5a1.s\*** **LAB4\_countingvowels.s**

```

1 ;The semicolon is used to lead an inline documentation
2 ;Please fill the following before exam
3 ;;;Your Name: KARTHIK. B ; AADITI. C; KARTHIK YT; ARAVIND K, HARSH G
4 ;;;Student Number: 2021BEC0034, 2021BEC0036, 2021BEC0037, 2021BEC0035, 2021BEC0043
5 ;;;Program question: Sum of the elements of the array if it is greater than 5.
6 ;;; Directives
7     PRESERVE8
8     THUMB
9
10 ; Vector Table Mapped to Address 0 at Reset
11 ; Linker requires __Vectors to be exported
12
13     AREA RESET, DATA, READONLY
14     EXPORT __Vectors
15
16 __Vectors
17     DCD 0x20001000 ; stack pointer value when stack is empty
18     DCD Reset_Handler ; reset vector
19
20     ALIGN
21
22 ;Your Data section
23 ;AREA DATA
24 SUMP DCD SUM
25 N DCD 7
26 NUM1 DCD 3,-7,2,-2,10,20,30
27 POINTER DCD NUM1
28     AREA MYRAM,DATA,READWRITE
29 SUM DCD 0
30 ; The program
31 ; Linker requires Reset_Handler
32
33     AREA MYCODE, CODE, READONLY
34
35     ENTRY
36     EXPORT Reset_Handler
37
38 Reset_Handler
39
40

```

**Registers**

Register	Value
<b>Core</b>	
R0	0x0000003C
R1	0x00000000
R2	0x0000002C
R3	0x0000001E
R4	0x10000000
R5	0x00000000
R6	0x0000003C
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R...	0x20001000
R...	0xFFFFFFFF
R...	0x00000052
xP...	0x61000000
<b>Banked</b>	
<b>System</b>	
<b>Internal</b>	
M...	Thread
Pri...	Privileged
St...	MSP
St...	3217813
Sec	0.26815108

**lab5a1.s\*** **LAB4\_countingvowels.s**

```

20     ALIGN
21
22 ;Your Data section
23 ;AREA DATA
24 SUMP DCD SUM
25 N DCD 7
26 NUM1 DCD 3,-7,2,-2,10,20,30
27 POINTER DCD NUM1
28     AREA MYRAM,DATA,READWRITE
29 SUM DCD 0
30 ; The program
31 ; Linker requires Reset_Handler
32
33     AREA MYCODE, CODE, READONLY
34
35     ENTRY
36     EXPORT Reset_Handler
37
38 Reset_Handler
39
40
41 ;;;;;;;;;;User Code Start from the next line;;;;;;;;;
42     LDR R1, N ; load size of array - a counter for how many elements are left to process
43     LDR R2, POINTER ; load base pointer of array
44     MOV R0, #0 ; initialize accumulator
45 LOOP
46     LDR R3, [R2], #4 ; load value from array, increment array pointer to next word
47     CMP R3, #5 ; compare loaded value with 5
48     BLE CHECK_NEXT ; if value is less than or equal to 5, skip adding
49     ADD R0, R0, R3 ; add value from array to accumulator
50 CHECK_NEXT
51     SUBS R1, R1, #1 ; decrement work counter
52     BGT LOOP ; keep looping until counter is zero
53     LDR R4, SUMP ; get memory address to store sum
54     STR R0, [R4] ; store answer
55     LDR R6, [R4] ; Check the value in the SUM
56 STOP
57     B STOP
58     END

```

## 5(b) – Find minimum and maximum element of given array.

**Registers**

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000044
R3	0xFFFFFFFF9
R4	0x00000042
R5	0x00000042
R6	0x0000000C
R7	0x00000008
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R...	0x20001000
R...	0xFFFFFFFF
R...	0x00000078
xP...	0x61000000

Banked  
System  
Internal

M... Thread  
Pri... Privileged  
St... MSP  
St... 3300778  
Sec 0.27506483

**lab5a1.s\***

```

1 ;The semicolon is used to lead an inline documentation
2 ;Please fill the following before exam
3 ;;;Your Name: KARTHIK. B ; AADITI. C; KARTHIK YT; ARAVIND K, HARSH G
4 ;;;Student Number: 2021BEC0034, 2021BEC0036, 2021BEC0037, 2021BEC0035, 2021BEC0043
5 ;;;Program Question: Finding Minimum and Maximum of the Array.
6 ;;; Directives
7 PRESERVE8
8 THUMB
9
10 ; Vector Table Mapped to Address 0 at Reset
11 ; Linker requires __Vectors to be exported
12
13 AREA RESET, DATA, READONLY
14 EXPORT __Vectors
15
16 __Vectors
17 DCD 0x20001000 ; stack pointer value when stack is empty
18 DCD Reset_Handler ; reset vector
19
20 ALIGN
21
22 ;Your Data section
23 ;AREA DATA
24 ;Max DCD 0
25 MaxP DCD Max
26 ;Min DCD 0
27 MinP DCD Min
28 N DCD 12
29 NUM1 DCD 3, -7, 2, -2, 10, 20, 30, 15, 32, 8, 64, 66
30 POINTER DCD NUM1
31 AREA MYRAM, DATA, READWRITE
32 Max DCD 0
33 Min DCD 0
34 ; The program
35 ; Linker requires Reset_Handler
36
37 AREA MYCODE, CODE, READONLY
38
39 ENTRY
40 EXPORT Reset_Handler

```

**Registers**

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000044
R3	0xFFFFFFFF9
R4	0x00000042
R5	0x00000042
R6	0x0000000C
R7	0x00000008
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R...	0x20001000
R...	0xFFFFFFFF
R...	0x00000078
xP...	0x61000000

Banked  
System  
Internal

M... Thread  
Pri... Privileged  
St... MSP  
St... 3300778  
Sec 0.27506483

**lab5a1.s\***

```

33 Min DCD 0
34 ; The program
35 ; Linker requires Reset_Handler
36
37 AREA MYCODE, CODE, READONLY
38
39 ENTRY
40 EXPORT Reset_Handler
41
42 Reset_Handler
43
44 ;;;;;;;;;;User Code Start from the next line;;;;;;;;;
45 LDR R1, N ; load size of array - a counter for how many elements are left to process
46 LDR R2, POINTER ; load base pointer of array
47 MOV R3, #0 ; initialize min to a large value
48 MOV R4, #0 ; initialize max to a small value
49
50 LOOP_MIN_MAX
51 LDR R5, [R2], #4 ; load value from array, increment array pointer to next word
52 CMP R5, R4 ; compare loaded value with max
53 BGT UPDATE_MAX ; if value is greater than max, update max
54 CMP R5, R3 ; compare loaded value with min
55 BLT UPDATE_MIN ; if value is less than min, update min
56 B CHECK_NEXT_MIN_MAX ; check next element
57 UPDATE_MIN
58 MOV R3, R5 ; update min
59 B CHECK_NEXT_MIN_MAX ; check next element
60 UPDATE_MAX
61 MOV R4, R5 ; update max
62 CHECK_NEXT_MIN_MAX
63 SUBS R1, R1, #1 ; decrement work counter
64 BGT LOOP_MIN_MAX ; keep looping until counter is zero
65 LDR R6, =MinP ; load memory address to store min
66 STR R3, [R6] ; store min
67 LDR R7, =MaxP ; load memory address to store max
68 STR R4, [R7] ; store max
69 STOP
70 B STOP
71 END

```