ECE 322 EMBEDDED SYSTEM

Lab :-4

Name :- ARVIND KUMAR

Roll N:-2021BEC0035

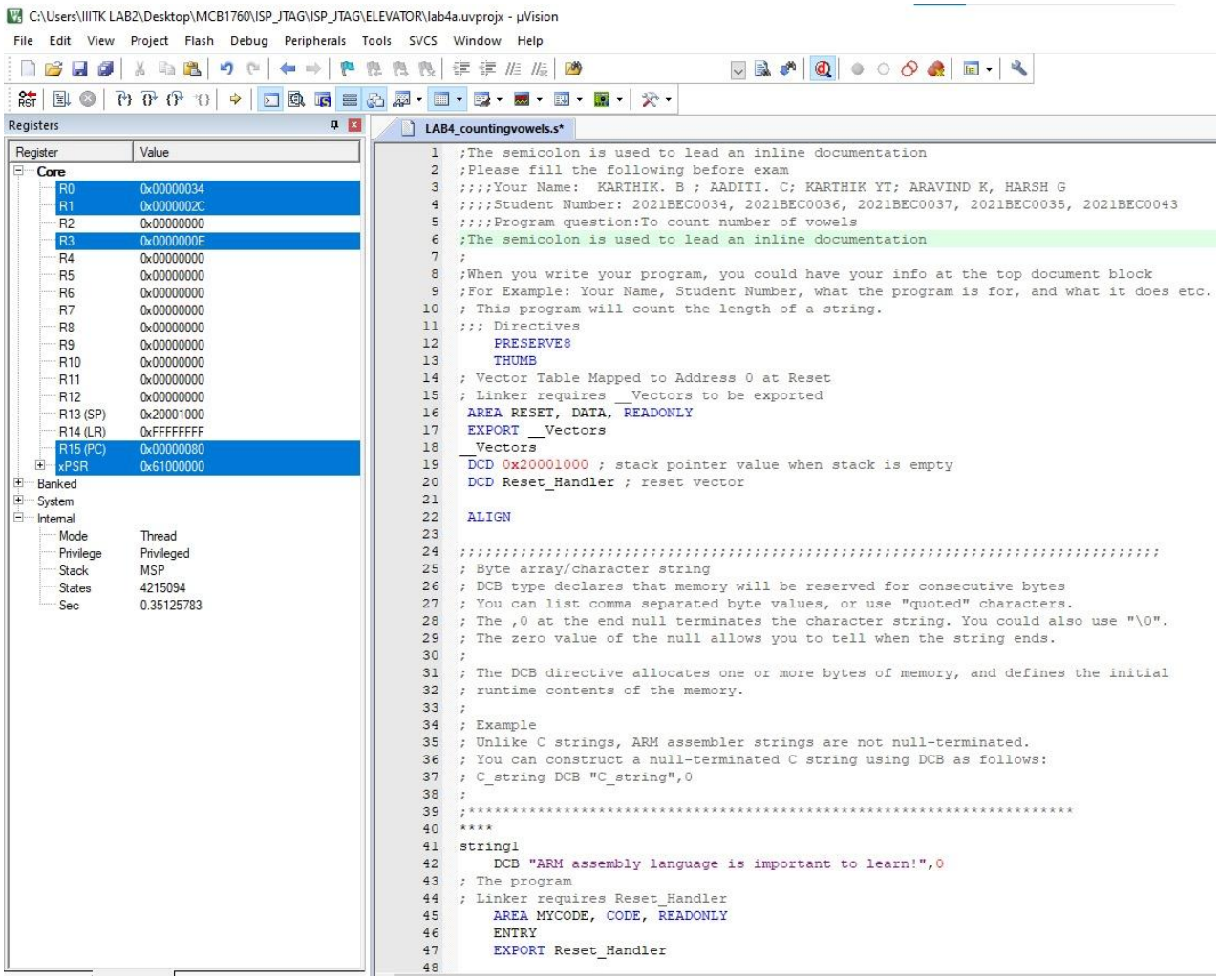
DATE : 23/01/2024

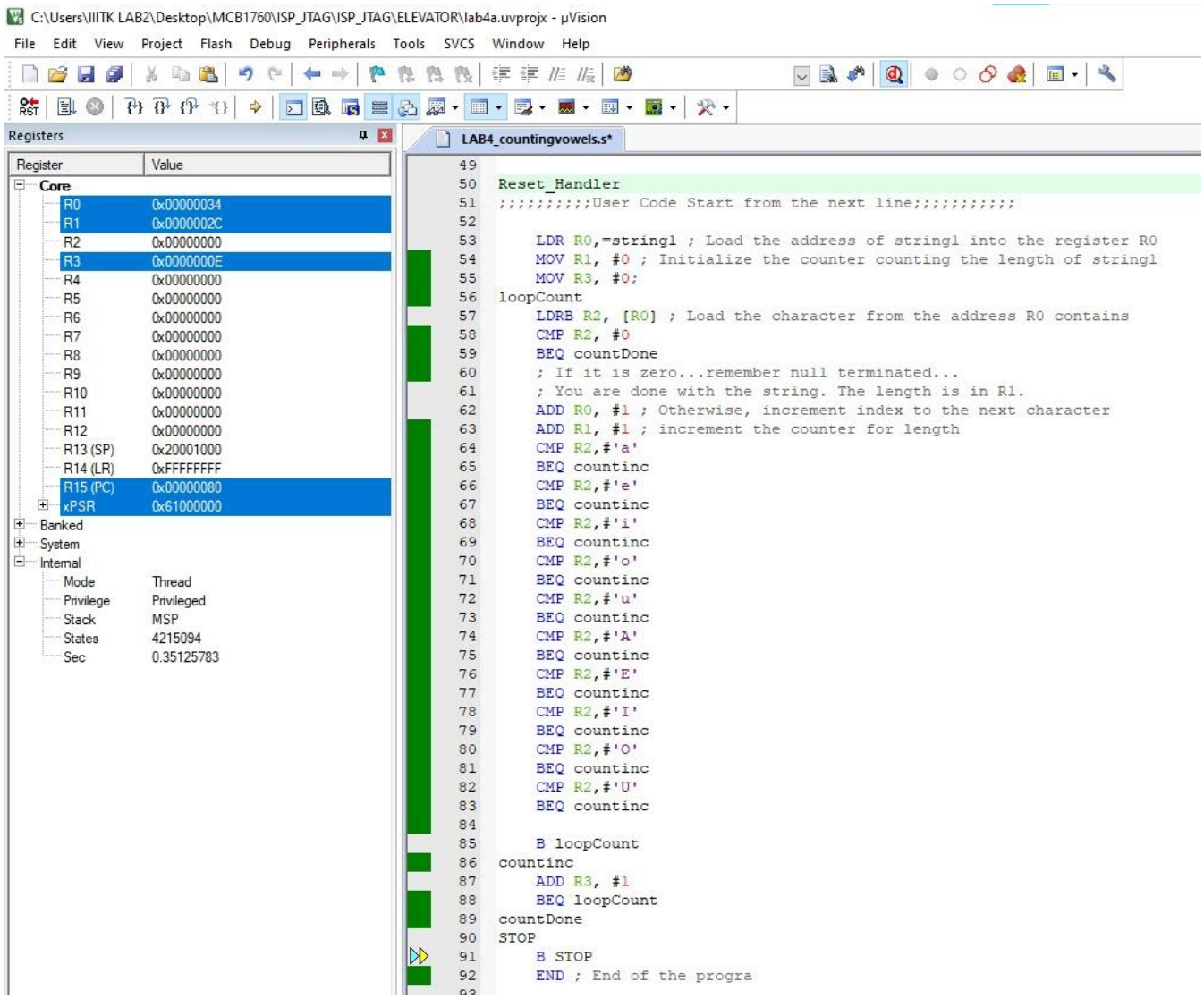
Submitted to - Della Ma'am

Aim:-

4(b) - Count the vowels in the given string = "ARM assembly language is important to learn!"

Code:


C:\Users\IIITK LAB2\Desktop\MCB1760\ISP_JTAG\ISP_JTAG\ELEVATOR\lab4a.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Core
R0 0x00000034
R1 0x0000002C
R2 0x00000000
R3 0x0000000E
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x20001000
R14 (LR) 0xFFFFFFFF
R15 (PC) 0x00000080
xPSR 0x61000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 4215094
Sec 0.35125783
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:To count number of vowels
6 ;The semicolon is used to lead an inline documentation
7 ;
8 ;When you write your program, you could have your info at the top document block
9 ;For Example: Your Name, Student Number, what the program is for, and what it does etc.
10 ; This program will count the length of a string.
11 ;;; Directives
12 PRESERVE8
13 THUMB
14 ; Vector Table Mapped to Address 0 at Reset
15 ; Linker requires __Vectors to be exported
16 AREA RESET, DATA, READONLY
17 EXPORT __Vectors
18 __Vectors
19 DCD 0x20001000 ; stack pointer value when stack is empty
20 DCD Reset_Handler ; reset vector
21
22 ALIGN
23
24 ;;
25 ; Byte array/character string
26 ; DCB type declares that memory will be reserved for consecutive bytes
27 ; You can list comma separated byte values, or use "quoted" characters.
28 ; The ,0 at the end null terminates the character string. You could also use "\0".
29 ; The zero value of the null allows you to tell when the string ends.
30 ;
31 ; The DCB directive allocates one or more bytes of memory, and defines the initial
32 ; runtime contents of the memory.
33 ;
34 ; Example
35 ; Unlike C strings, ARM assembler strings are not null-terminated.
36 ; You can construct a null-terminated C string using DCB as follows:
37 ; C_string DCB "C_string",0
38 ;
39 ;***
40 ****
41 string1
42 DCB "ARM assembly language is important to learn!",0
43 ; The program
44 ; Linker requires Reset_Handler
45 AREA MYCODE, CODE, READONLY
46 ENTRY
47 EXPORT Reset_Handler
48


C:\Users\IIITK LAB2\Desktop\MCB1760\ISP_JTAG\ISP_JTAG\ELEVATOR\lab4a.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Core
R0 0x00000034
R1 0x0000002C
R2 0x00000000
R3 0x0000000E
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x20001000
R14 (LR) 0xFFFFFFFF
R15 (PC) 0x00000080
xPSR 0x61000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 4215094
Sec 0.35125783
LAB4_countingvowels.s*
49
50 Reset_Handler
51 ;;;;;;;;;;User Code Start from the next line;;;;;;;;;;
52
53 LDR R0,=string1 ; Load the address of string1 into the register R0
54 MOV R1, #0 ; Initialize the counter counting the length of string1
55 MOV R3, #0;
56 loopCount
57 LDRB R2, [R0] ; Load the character from the address R0 contains
58 CMP R2, #0
59 BEQ countDone
60 ; If it is zero...remember null terminated...
61 ; You are done with the string. The length is in R1.
62 ADD R0, #1 ; Otherwise, increment index to the next character
63 ADD R1, #1 ; increment the counter for length
64 CMP R2,#'a'
65 BEQ countinc
66 CMP R2,#'e'
67 BEQ countinc
68 CMP R2,#'i'
69 BEQ countinc
70 CMP R2,#'o'
71 BEQ countinc
72 CMP R2,#'u'
73 BEQ countinc
74 CMP R2,#'A'
75 BEQ countinc
76 CMP R2,#'E'
77 BEQ countinc
78 CMP R2,#'I'
79 BEQ countinc
80 CMP R2,#'O'
81 BEQ countinc
82 CMP R2,#'U'
83 BEQ countinc
84
85 B loopCount
86 countinc
87 ADD R3, #1
88 BEQ loopCount
89 countDone
90 STOP
91 B STOP
92 END ; End of the progra
93