Question 1 For the assembly program given in Figure 1 do the following task, assuming that the text/code

section starts at 0x00000000 and data section starts at 0x00000050.

(a) Write down the definition (what it is?) and functionality (what it does and how) of each line of the program.

(b) Determine the content of registers used and memory for the first time when the instruction is being executed. (note that an instruction may get executed for multiple time, but the question is to note down whenever the change happens for the first time). Refer Table 1 for note your answer.

(c) Determine the content of the registers and memory for the last time when the instruction is being executed. Refer Table 1 to note your answer.

(d) Figure out the functionality of the program, what it is doing?

a)CODE EXPLANATION:

.data//define data section

tirupati://defining Tirupati array(word)

.word 0xe, 0xb, 0xa, 0xd, 0xe, 0xf, 0xa, 0xf, 0xc, 0xd

About the above code :In the above code we have the word Tirupati defined (initialised)by an array of numbers in hexadecimal under .data category ie: 0xe, 0xb, 0xa, 0xd, 0xe, 0xf, 0xa, 0xf, 0xc, 0xd

.text //to categorise this comes under text section

.global _start //declares the start of the program as global parameter(function)

_start://function name start

ldr r0, =Tirupati //r0 register loads the address of Tirupati word

ldr r1, =10 //r1 register is loaded with 10

mov r2, #0 //r2 register now contains 0 as value

yerpedu://function/subroutine yerpedu is defined

mov r3, #0 //register r3 now contains value 0

kalahasthi: //function/subroutine kalahasti is defined

ldr r4, [r0, r3, lsl #2]//r4 register is loaded with value in the address location r0+val in r3 left shifted by 2 places.ie:address in r0 alongwith offset r3 leftshift by 2

add r7, r3, #1 //register r7 has sum value of r3 and 1(r3+1)

ldr r5, [r0, r7, lsl #2] //r5 register is loaded with value in the address location r0+val in r7 left shifted by 2 places.ie:address in r0 alongwith offset r7 leftshift by 2

cmp r4, r5 //to compare values contained in r4 and r5 registers

bllt ulta //branch to ulta subroutine if value in r4 is less than value in r5

add r3, r3, #1 //add value in r3 with 1 and store it in r3

cmp r3, r1 //compare values in r3 and r1 registers

blt kalahasthi //if the value of above comparision leads to less than result then branch to kalahasti subroutine

cmp r2, r1//compare values in r2 and r1 registers

sub r1, r1, #1//subtracts value in r1 by 1

blt yerpedu //if the value of r2 register is lesser than r1 branch to yerpedu subroutine

end://define end subroutine

b end//branch to end subroutine infinite loop marking the end

ulta://define ulta function/subroutine

str r5, [r0, r3, lsl #2]// value in r5 register is stored to address location r0+val in r3 left shifted by 2 places.ie:address in r0 alongwith offset r3 leftshift by 2

str r4, [r0, r7, lsl #2] // value in r4 register is stored to address location r0+val in r7 left shifted by 2 places.ie:address in r0 alongwith offset r7 leftshift by 2

bx lr //return the link register to the last call of the subroutine

.end //end defined

(B,C)Table 1: All the content for First Time and Last Time are to be written in this table

	First Time	Last Time
PC (R15)		
LR (R14)		
R0	0x00000050	0x00000078

R1	10	-1
R2	0	0
R3	0	1
R4	0xe	0xf
R5	0xb	0xf
R7	1	1

d)this program does bubble sort algorithm by sorting adjacent elements and arrange them in descending order in the memory.