Problem

For the below code, if the label "YOU" is at address x4005, what are the addresses of the labels THEY, WE and ALL? (10 Points)

YOU .FILL xF010

THEY .BLKW 10

WE .STRINGZ "MSCS3070"

ALL .FILL x0005

Solutions: YOU: x4005 THEY: x4006 WE:x4012 ALL: x4016

* Write are the ranges of offsets for the BR and JSR instructions (5 Points)

BR: [-256, +255]

JSR: [-1024, +1023]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem

What values are in the value array at end of program?

int value[5];

int\* ptr;

void main() {

int i;

ptr = &value;

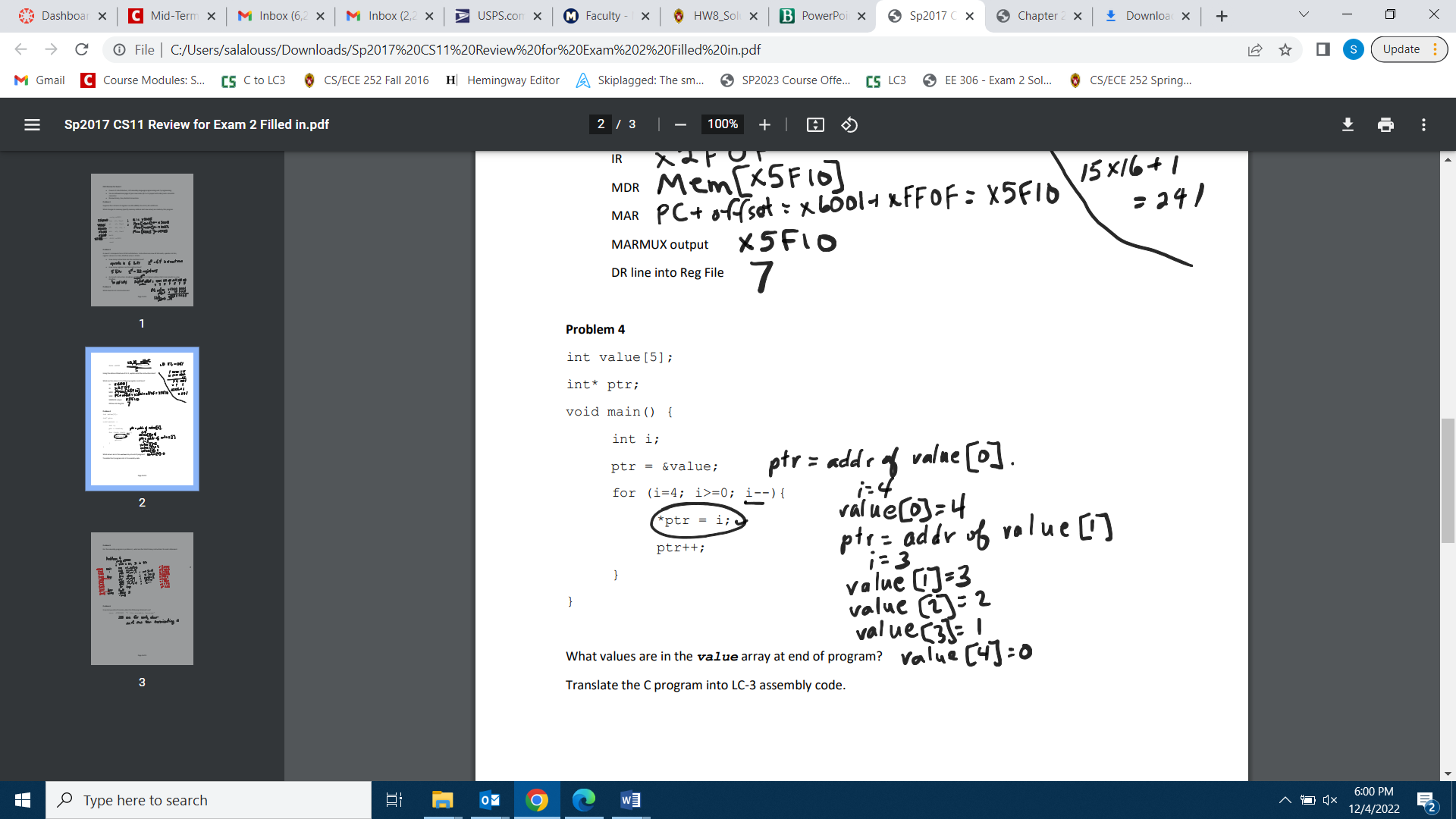
for (i=4; i>=0; i--){

\*ptr = i;

ptr++;

}

}



\_\_\_\_\_\_\_\_\_\_\_

Problem

How many words of memory does the following statement use? Explain in details

vars .STRINGZ “My Clark University”

19 one for each char and one for terminating 0.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem

1. Briefly explain the difference between *interrupt-driven I/O* and *polling based*

*I/O*? (2 Points)

Polling: CPU keeps checking status register until new data arrives or device ready for new data.

Interrupt: Device sends a special signal to CPU when new data arrives or device ready for next data.

1. What is the main reason to prefer asynchronous I/O over synchronous I/O in recent microprocessor designs? (2 Point)

I/O devices usually operate at speeds very different from that of a microprocessor. The rate at which data is provided or consumed is not predictable and usually not in lockstep with the processor clock.

1. How many trap service routines can be defined in LC-3? (2 Point)

256

1. Explain briefly the problem that the *callee-save* and the *caller-save* approaches are trying to solve. (2 Point)

If a register value is “destroyed” by actions of a subroutine or service routine, the value has to be saved before it is modified, and reloaded before it is used again.

1. What important feature does the instruction JSRR provide that JSR does not?

(2 Points)

JSRR uses the contents a register as the address to jump to (16 bits), while JSR instruction provides an 11 bit offset to PC. Thus the range of addresses to which a JSRR instruction can jump to is larger than that of the JSR instruction.

1. What happens during the linkingphase of an assembly program? (2 Points)

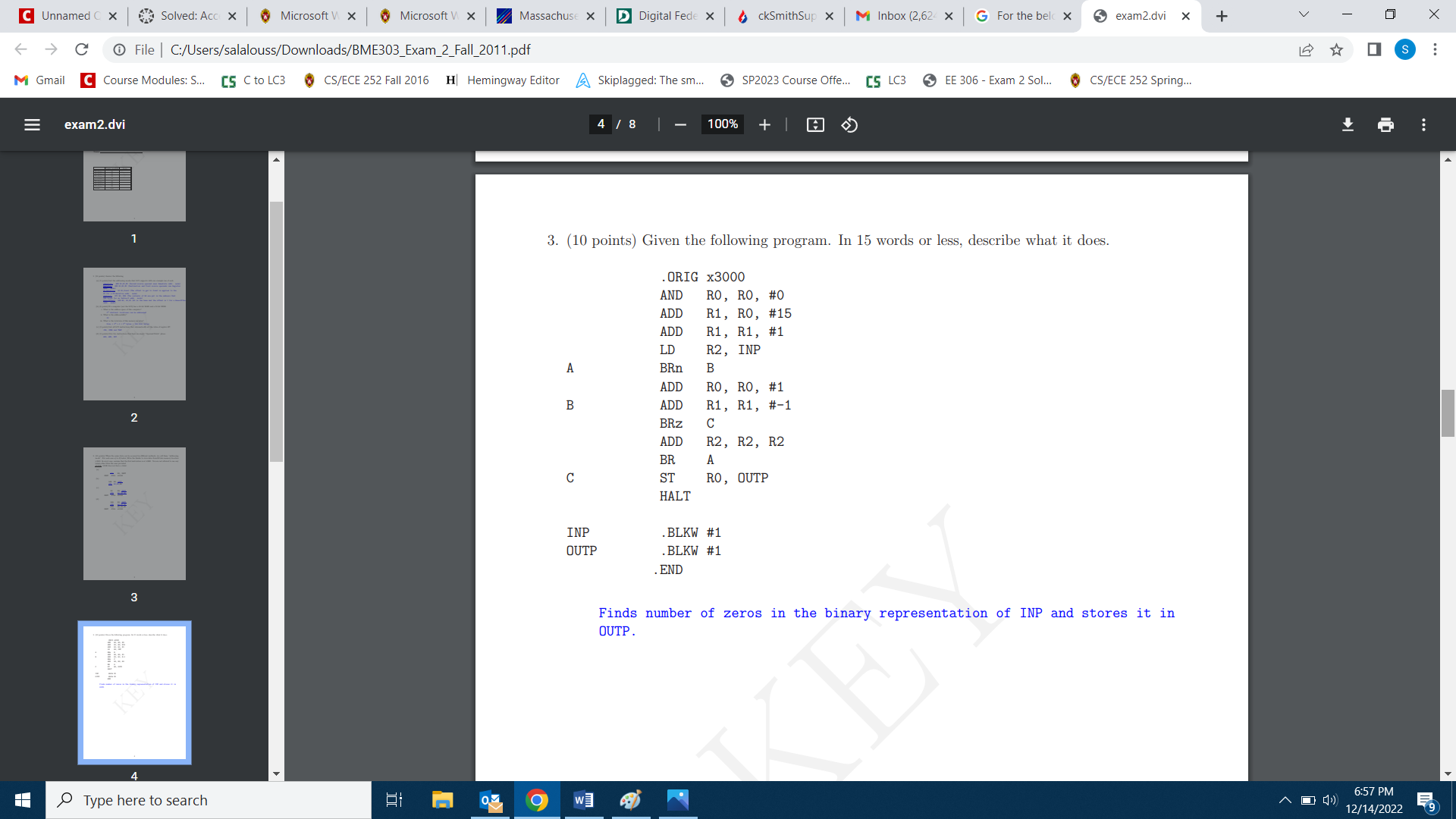
Linking is the process of resolving symbols between independent object files. The linker will search symbol tables of other modules to resolve symbols and complete code generation before loading.

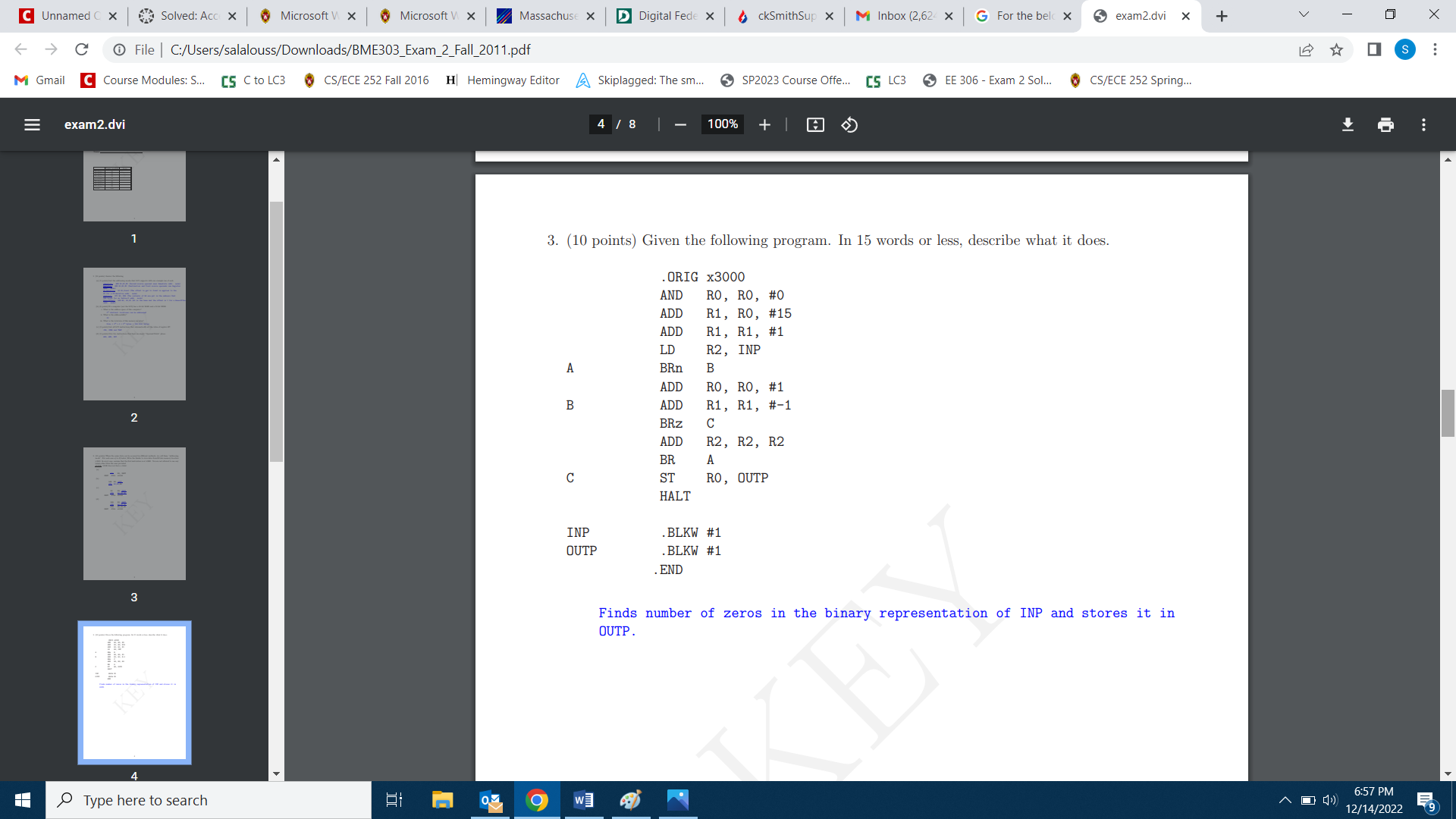
1. What is the use of *Comments* in an assembly program? (2 Points)

Comments are useful to humans to document or understand programs. They are ignored by the assembler.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem:

An LC-3 assembly language program in given below:

.ORIG x3000

AND R3, R3, #0

LD R0, M0

LD R1, M1

LD R2, M2

ADD R3, R3, R2

LOOP ADD R3, R3, #1

ADD R0, R0, #-1

BRn LOOP

DONE ST R3, RESULT

HALT

RESULT .FILL x0000

M0 .BLKW #7

M1 .STRINGZ “Section-3”

M2 .FILL x0009 .END

1. A symbol table is created during the first pass by the assembler. Fill in the

following symbol table for the above program: (12 Points)

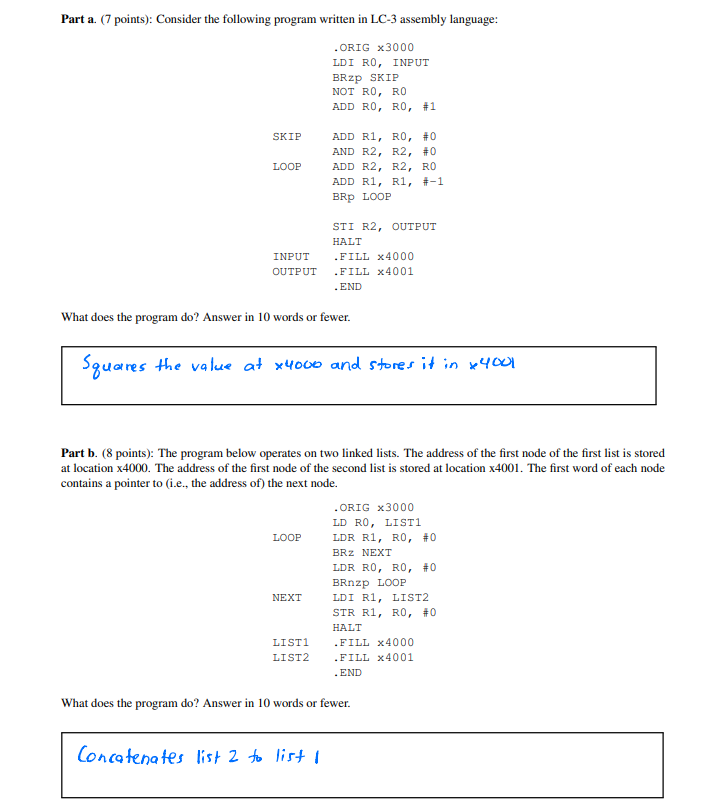
|  |  |
| --- | --- |
| Symbol | Address |
| LOOP  DONE  RESULT M0  M1  M2 | x3005 x3008  x300A x300B x3012  x301C |

1. The assembly program is converted into a binary file during the second pass by the assembler. Fill in the binary instructions at the following memory locations: (8 Points)

|  |  |
| --- | --- |
| Address | Instructions |
| x3007 | 0000 1001 1111 1101 |
| x3008 | 0011 0110 0000 0001 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem

Suppose the following LC-3 subroutine implements a new service routine called GETS. The subroutine will store the input string starting at the address in R0 and then return to normal execution. It performs this operation by repeatedly taking input characters from the keyboard and storing it in the location specified by R0 until it sees the '\n' character. Note: The most significant bit of the KBSR is 1 if keyboard has received a new character.

1. Fill in the blanks. There should be only one instruction per line. (4 Points)

.ORIG x0650  
ST R0, R0\_TMP  
ST R1, R1\_TMP  
ST R2, R2\_TMP

L1      LDI R1, KBSR  
(a)      BRzp L1 ; Check KBSR  
(b)      LDI R2, KBDR ; Load value in the KBDR into R2  
          LD R1, NEGCHAR  
          ADD R1, R1, R2  
          BRz DONE ; Check for '\n'  
          STR R2, R0, #0  
          ADD R0, R0, #1  
         BRnzp L1

DONE (c) AND R2, R2, #0  
         STR R2, R0, #0 ; Store NULL CHAR  
         LD R2, R2\_TMP  
         LD R1, R1\_TMP  
         LD R0, R0\_TMP  
(d)    RET  
KBSR .FILL xFE00 ; Address of KBSR  
KBDR .FILL xFE02 ; Address of KBDR  
NEGCHAR .FILL xFFF6 ; Negative value of character '\n'  
R0\_TMP .FILL 0  
R1\_TMP .FILL 0  
R2\_TMP .FILL 0  
         .END

Assume the above assembly code is a service routine that can be called using TRAP x55. What is the address of the corresponding System Control Block entry and what are its contents? Give your answer in hex.

|  |  |
| --- | --- |
| Address of trap vector table entry | Contents at this memory location |
| 0x0055 | 0x0650 |