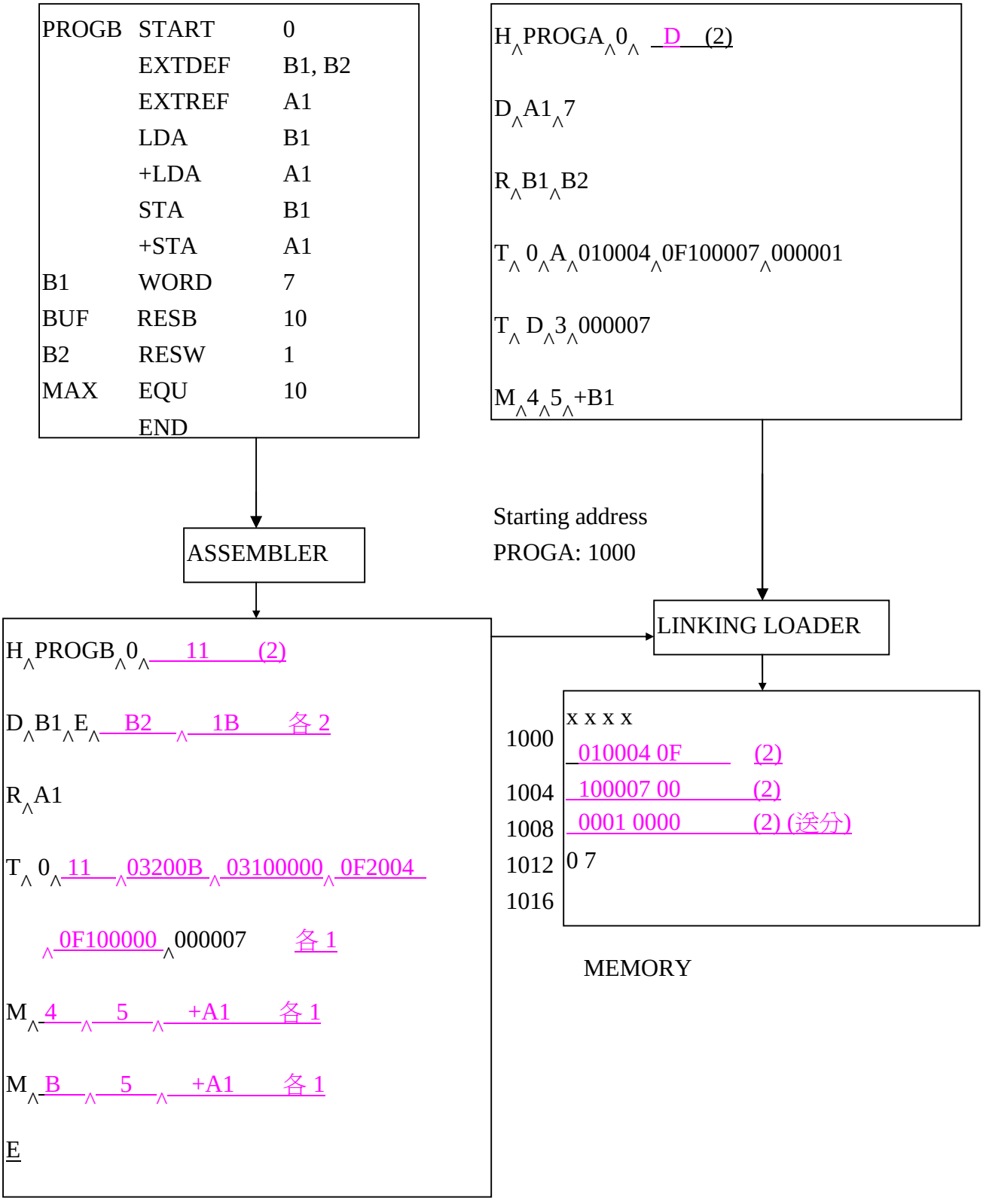
國立嘉義大學資訊工程學系系統程式期中考考卷

學號：　　姓名：

1. Please write the content in the lines. (25%)



NOTE: 1. The opcodes of LDA and STA are “00” and “0C”, respectively.
2. PROGB is placed after ROGA.
3. Each x represents 4 bits.

2. Please explain the following keywords： (6%)
   (a) Reference number
   (b) External reference

(a)assigning a reference number to each external symbol referred to in a control section

- Control section name: 01
- Other external reference symbols (stored in the Refer records): 02symname, 03symname, …

– using this reference number (instead of the symbol name) in Modification records

(b) symbols that are used in this control section and are defined elsewhere

3. Please explain the difference between “program block” and “control section”. (6%)

Program blocks

Segments of code that are rearranged within a single object program unit

Control sections

Segments of code that are translated into independent object program units

4. Please describe how to use “relocation bit” for SIC programs. (5%)

- When the instruction format is fixed as in SIC machine (one word per instruction), we can associate each instruction with a relocation bit.
- Relocation bits can be gathered together into a bit mask following the length indicator in each Text record.
- If bit=1, the corresponding word of object code is relocated.

5. Please show how to compute the target addresses and values in register A of the following SIC/XE machine codes. And complete the two (20%) 各 2

I:003030	003600
J:003600	103000
K:006390	00C303
L:00C303	003030

(B) =006000

(PC)=003000

(X) =000090

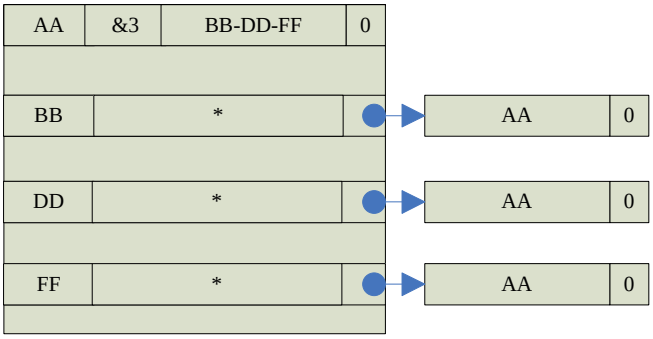
Assembler language	HEX	$(TA)_{16}$	(Value in A)$_{16}$
	032600	3600	103000
	022030	3030	103000
	036000	X	X
+LDA L	0310C303		003030
LDA @K	024390		003030

(hint: the opcode of LDA is 00.)

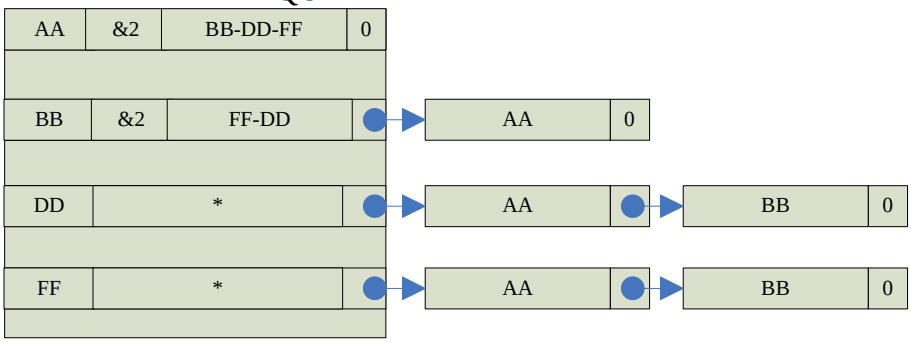
4. Please show the content of the symbol table when instructions are read in Multi-Pass assemblers. (依字母由 A 到 D)?(12%)

```
AA      EQU     BB – DD - FF
BB      EQU     FF - DD
FF      EQU     4096
DD      EQU     256
```

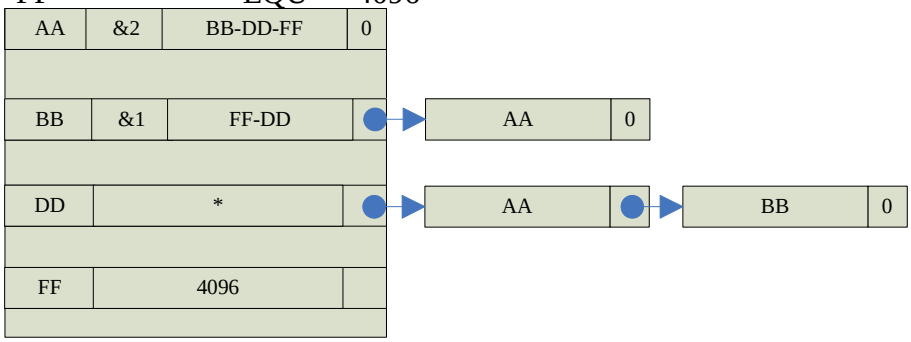
AA EQU BB – DD - FF



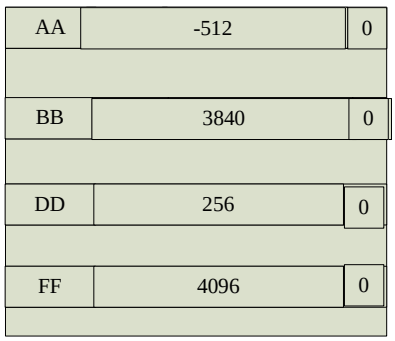
BB EQU FF - DD



FF EQU 4096



DD EQU 256

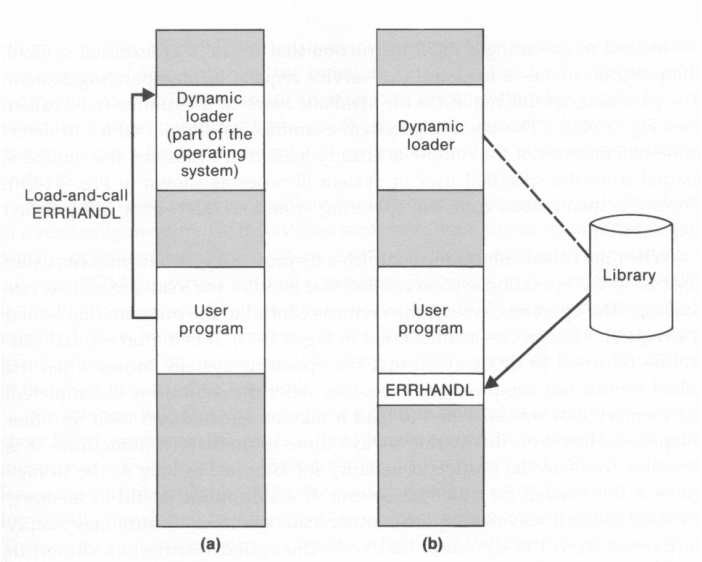


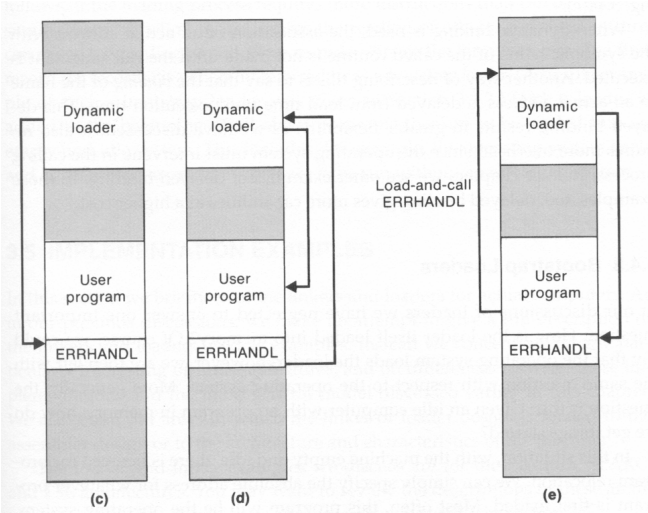
5. What is “bootstrap loader”? (5%)

- An absolute loader program is permanently resident in a read-only memory (ROM)
- Copy absolute loader in ROM into RAM for execution (optional)
- Read a fixed-length record from some device into memory at a fixed location. After the read operation, control is automatically transferred to the address in memory where the record was stored
  - The record contains instructions that load the following absolute program
    - Cause reading of other records.

6. Please describe the loading and calling of a subroutine using dynamic linking?

8%





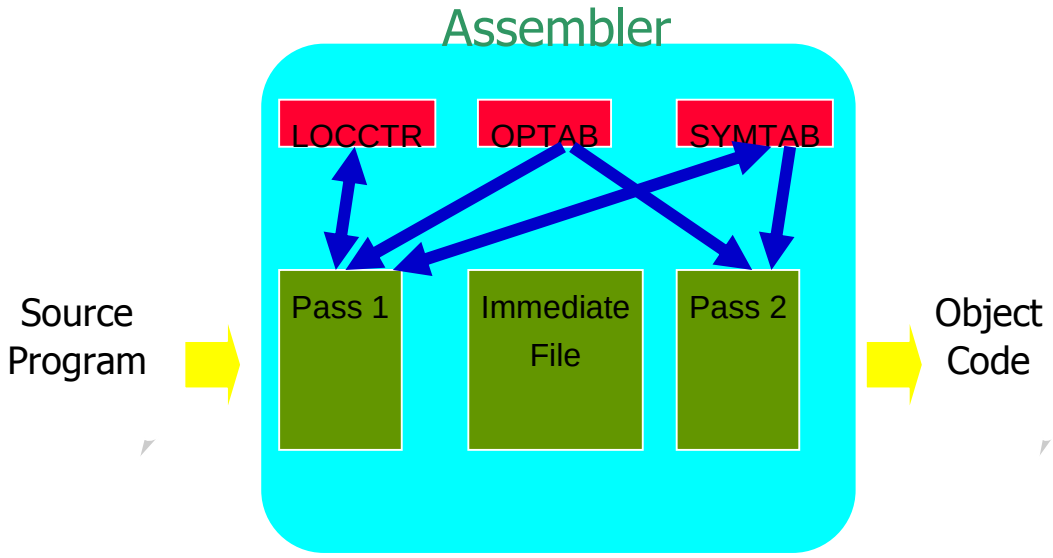
7. When do linkage editor, linking loader, dynamic linking loaders perform these same linking operations, respectively? (6%)

Linkage editors: perform these same (linking) operations before load time
Linking loaders: perform these same (linking) operations at load time
Dynamic linking: perform linking at execution time

8. Please describe the behaviors of an assembler in Pass1 and Pass2. What data structures do we need for such behaviors? (8%)

- Pass 1 (define symbols) 3
  - Assign addresses to all statements in the program
  - Save the addresses assigned to all labels for use in Pass 2
  - Perform assembler directives, including those for address assignment, such as BYTE and RESW
- Pass 2 (assemble instructions and generate object program) 3
  - Assemble instructions (generate opcode and look up addresses)
  - Generate data values defined by BYTE, WORD
  - Perform processing of assembler directives not done during Pass 1
  - Write the object program and the assembly listing
- Two basic internal data structures: 2
  - The Operation Code Table (OPTAB)
    - Look up mnemonic operation codes
    - Translate the codes to the corresponding machine language equivalents
  - The Symbol Table (SYMTAB)
    - Store addresses assigned to labels

- Location Counter (LOCCTR)
  - Initialized to the beginning address specified in the START statement
  - A variable for maintaining the assignment of addresses
  - When a label is reached in the program, the current value of LOCCTR is associated with the label.



9. Please complete the object program according to one-pass assembler. (8%)

H^ COPY ^00100000107A
T^001000^09^454F46^000003^000000
T^00200F^15^141009^480000^00100C^281006^300000^480000^3C2012
T^__00201C__^02^__2024_____
T^002024^19^001000^0C100F^001003^0C100C^480000^081009^4C0000^F1^001000
T^__002013____^02^__203D_____
T^00203D^1E^041006^001006^E02039^302043^D82039^281006^300000^54900F^2C203A^382043

9. Please explain why the operands of the three instructions in the following program are different. (6%)

(1) LDB #LENGTH
BASE LENGTH

(2) LDA LENGTH

(3) LDA #3

Loc		Source statement		Object code
0000	COPY	START	0	
0000	FIRST	STL	RETADR	17202D
0003		LDB	#LENGTH	69202D
		BASE	LENGTH	
0006	CLOOP	+JSUB	RDREC	4B101036
000A		LDA	LENGTH	032026
000D		COMP	#0	290000
0010		JEQ	ENDFIL	332007
0013		+JSUB	WRREC	4B10105D
0017		J	CLOOP	3F2FEC
001A	ENDFIL	LDA	EOF	032010
001D		STA	BUFFER	0F2016
0020		LDA	#3	010003
0023		STA	LENGTH	0F200D
0026		+JSUB	WRREC	4B10105D
002A		J	@RETADR	3E2003
002D	EOF	BYTE	C'EOF'	454F46
0030	RETADR	RESW	1	
0033	LENGTH	RESW	1	
0036	BUFFER	RESB	4096	

- LDB #LENGTH: loads this value into register B during program execution.
  BASE LENGTH: informs the assembler that the base register will contain the address of LENGTH.
- PC-relative addressing
- Use immediate mode. The target address becomes the operand

10. 請問你對這門課下半學期的教學有何建議? (5%)