

考生姓名：

學號：

得分：

以下題目均按學號規則推算出指定的題數來做，學號為奇數者做奇數題，偶數者做偶數題，做錯題目則該題不計分，共 99 分，1 分奉送！

名詞解釋：每題 3 分，做指定的 4 題，共 12 分。

1. 系統程式
2. 系統軟體
3. gcc
4. objdump
5. ar
6. make
7. cpu0
8. ORG 假指令

簡答題：每題 5 分，做指定的 4 題，共 20 分。

1. 請畫出電腦的馮紐曼架構，並說明其中各個元件的功用？
2. 請說明組譯器第一階段的功能？
3. 請說明組譯器第二階段的功能？
4. 請說明指令表在組譯器中的功能？
5. 請說明符號表在組譯器中的功能？
6. 請說明連結器的功能為何？
7. 請說明動態連結器的功能為何？
8. 請說明載入器的功能為何？

考生姓名：

學號：

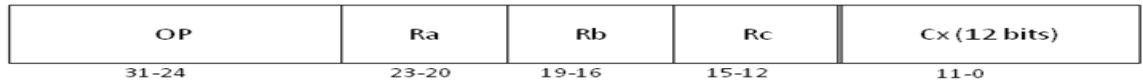
程式題：每題 10 分，做指定的 4 題，共 40 分。

- |                                                                                                                                                                            |                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1. 請寫出一個組合語言程式，可以將變數 <b>Y</b> 的內容平方後放到變數 <b>Y2</b> 中。</li><li>2. 請寫出一個組合語言程式，可以將變數 <b>X</b> 的內容加 <b>10</b> 後放回 <b>X</b> 中。</li></ol> | <ol style="list-style-type: none"><li>3. 請寫出一個組合語言程式，可以將變數 <b>X, Y</b> 的內容交換。</li><li>4. 請寫出一個組合語言程式，可以取得變數 <b>X, Y</b> 中較大的值放入變數 <b>max</b> 中。</li></ol>                          |
| <ol style="list-style-type: none"><li>5. 請寫出一個「組合語言副程式」，可以將 <b>R2, R3</b> 中較小者放入 <b>R1</b> 中。</li><li>6. 請寫出一個「組合語言副程式」，可以設定 <b>R1</b> 為 <math>3^{R2}</math>。</li></ol>    | <ol style="list-style-type: none"><li>7. 請寫出一個組合語言程式，給定整數陣列 <b>A</b> 可以計算出 <b>A</b> 內容的總和放在 <b>SUM</b> 變數中。</li><li>8. 請寫出一個組合語言程式，給定字串 <b>A</b>，可以將該字串複製到字串 <b>B</b> 中。</li></ol> |

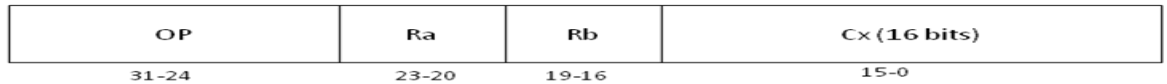
請寫出下列程式的位址欄與機器碼 (每列 3 分，位址欄 1 分，機器碼 2 分，做指定的 9 列，共 27 分)

題號	位址欄	組合語言	機器碼
1.		LDI R1, 1	
2.		LD R2, aptr	
3.		LD R3, bptr	
4.		LD R4, len	
5.		ADD R3, R3, R4	
6.		FOR: CMP R3, R0	
7.		JLT EXIT	
8.		LD R5, [R1]	
9.		ST R5, [R2]	
10.		ADD R2, R2, R1	
11.		SUB R3, R3, R1	
12.		JMP FOR	
13.		EXIT: RET	
14.		len: WORD 6	
15.		a: BYTE 3, 2, 7, 6, 8, 5	
16.		b: RESB 6	
17.		aptr: WORD a	
18.		bptr: WORD b	

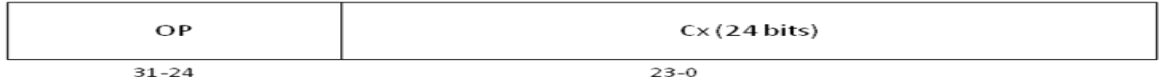
## A 型



## L 型



## J 型



類型	格式	指令	OP	說明	語法	語意
載入 儲存	L	LD	00	載入 word	LD Ra, [Rb+Cx]	$Ra \leftarrow [Rb + Cx]$
	L	ST	01	儲存 word	ST Ra, [Rb+ Cx]	$Ra \rightarrow [Rb + Cx]$
	L	LDB	02	載入 byte	LDB Ra, [Rb+ Cx]	$Ra \leftarrow (\text{byte})[Rb + Cx]$
	L	STB	03	儲存 byte	STB Ra, [Rb+ Cx]	$Ra \rightarrow (\text{byte})[Rb + Cx]$
	A	LDR	04	LD 的暫存器版	LDR Ra, [Rb+Rc]	$Ra \rightarrow (\text{byte})[Rb + Rc]$
	A	STR	05	ST 的暫存器版	STR Ra, [Rb+Rc]	$Ra \rightarrow [Rb + Rc]$
	A	LBR	06	LDB 的暫存器版	LBR Ra, [Rb+Rc]	$Ra \leftarrow (\text{byte})[Rb + Rc]$
	A	SBR	07	STB 的暫存器版	SBR Ra, [Rb+Rc]	$Ra \rightarrow (\text{byte})[Rb + Rc]$
	L	LDI	08	立即載入	LDI Ra, Rb+Cx	$Ra \leftarrow Rb + Cx$
運算 指令	A	CMP	10	比較	CMP Ra, Rb	$SW \leftarrow Ra \geq Rb$
	A	MOV	12	移動	MOV Ra, Rb	$Ra \leftarrow Rb$
	A	ADD	13	加法	ADD Ra, Rb, Rc	$Ra \leftarrow Rb + Rc$
	A	SUB	14	減法	SUB Ra, Rb, Rc	$Ra \leftarrow Rb - Rc$
	A	MUL	15	乘法	MUL Ra, Rb, Rc	$Ra \leftarrow Rb * Rc$
	A	DIV	16	除法	DIV Ra, Rb, Rc	$Ra \leftarrow Rb / Rc$
	A	AND	18	邏輯 AND	AND Ra, Rb, Rc	$Ra \leftarrow Rb \text{ and } Rc$
	A	OR	19	邏輯 OR	OR Ra, Rb, Rc	$Ra \leftarrow Rb \text{ or } Rc$
	A	XOR	1A	邏輯 XOR	XOR Ra, Rb, Rc	$Ra \leftarrow Rb \text{ xor } Rc$
	A	ROL	1C	向左旋轉	ROL Ra, Rb, Cx	$Ra \leftarrow Rb \text{ rol } Cx$
	A	ROR	1D	向右旋轉	ROR Ra, Rb, Cx	$Ra \leftarrow Rb \text{ ror } Cx$
	A	SHL	1E	向左移位	SHL Ra, Rb, Cx	$Ra \leftarrow Rb \ll Cx$
	A	SHR	1F	向右移位	SHR Ra, Rb, Cx	$Ra \leftarrow Rb \gg Cx$
跳躍 指令	J	JEQ	20	跳躍 (相等)	JEQ Cx	if $SW(=)$ $PC \leftarrow PC + Cx$
	J	JNE	21	跳躍 (不相等)	JNE Cx	if $SW(!=)$ $PC \leftarrow PC + Cx$
	J	JLT	22	跳躍 (<)	JLT Cx	if $SW(<)$ $PC \leftarrow PC + Cx$
	J	JGT	23	跳躍 (>)	JGT Cx	If $SW(>)$ $PC \leftarrow PC + Cx$
	J	JLE	24	跳躍 (<=)	JLE Cx	if $SW(<=)$ $PC \leftarrow PC + Cx$
	J	JGE	25	跳躍 (>=)	JGE Cx	If $SW(>=)$ $PC \leftarrow PC + Cx$
	J	JMP	26	跳躍 (無條件)	JMP Cx	$PC \leftarrow PC + Cx$
	J	SWI	2A	軟體中斷	SWI Cx	$LR \leftarrow PC; PC \leftarrow Cx$
	J	CALL	2B	跳到副程式	CALL Cx	$LR \leftarrow PC; PC \leftarrow PC + Cx$
J	RET	2C	返回	RET	$PC \leftarrow LR$	
堆疊 指令	A	PUSH	30	推入 word	PUSH Ra	$SP -= 4; [SP] = Ra;$
	A	POP	31	彈出 word	POP Ra	$Ra = [SP]; SP += 4;$
	A	PUSHB	32	推入 byte	PUSHB Ra	$SP--; [SP] = Ra; (\text{byte})$
	A	POPB	33	彈出 byte	POPB Ra	$Ra = [SP]; SP++; (\text{byte})$