Started on	[redacted]
State	
Completed on	
Time taken	
Marks	
Grade	
Feedback	

You are a highly motivated student, who takes full responsibility for your learning. A reflective learner, who recognises areas for development and is committed to personal improvement. An organised learner who always completes class work and homework to a very high standard.

Question **1**

Correct

Mark 1.00 out of 1.00

In assembly language, what is the meaning of the mnemonic, BRLT?

a. Branch if limit threshold is x

b. Branch if less than x ✔

c. Branch if larger than x

d. Branch if equal to x

Your answer is correct.

The correct answer is:
Branch if less than x

Question **2**

Correct

Mark 1.00 out of 1.00

Which of the following mnemonics is a key component used to simulate a loop structure in assembly code?

Select one:

- a. STA
- b. ADDA
- c. LPA
- d. LOOPA
- e. LDA
- f. DECI
- g. SUBA
- h. BR ✔

Your answer is correct.

The correct answer is: BR

Question **3**

Correct

Mark 1.00 out of 1.00

What is the purpose of Assembly language?

- a. To create high-level abstractions of program logic
- b. To perform database queries
- c. To design graphical user interfaces (GUIs)
- d. To write software applications for specific platforms or devices ✔

Your answer is correct.

The correct answer is:
To write software applications for specific platforms or devices

Question **4**

Correct

Mark 2.00 out of 2.00

How can the Instruction Format in machine language be broken down, and what are the two components?

Select one:

- a. Instruction operation and operation code
- b. Instruction specifier and operation specifier ✔
- c. Instruction specifier and register specifier
- d. Addressing mode and operation code

Your answer is correct.

The correct answer is: Instruction specifier and operation specifier

Question **5**

Correct

Mark 1.00 out of 1.00

What is the purpose of the addressing-mode specifier in the instruction format?

- a. It contains the address of the next instruction to be executed.
- b. It specifies which register is to be used.
- c. It specifies how to interpret the operand part of the instruction. ✔
- d. It specifies which instruction is to be carried out.
- e. It contains a copy of the instruction being executed.

Your answer is correct.

The correct answer is:
It specifies how to interpret the operand part of the instruction.

Question **6**

Correct

Mark 22.00 out of 22.00

Suppose that you have an A-register (accumulator) of 16 bits; an instruction register of 8 bits; an operand specifier of 16 bits; and a program counter of 16 bits. Assume that memory has been reset and each cell of memory is holding 8-bits. Consider the listing trace of an assembler program below. From left to right it shows the: memory address/location, machine code, assembler instruction, and a comment (to remind you of the meaning of the assembler mnemonic).

Memory Location	Machine Code	Assembler Code	Comment	
0000	040017	**br** main	;**br** = branch	(0)
0003	0000	number: *.block 2*		(1)
0005	0000	factor: *.block 2*		
0007		pr: *.ASCII* "prime\x00"		
000D		npr: *.ASCII* "not prime\x00"		
0017	310003	main: **deci** number,d	;**deci** = read in number in decimal	(2)
001A	C10003	**lda** number,d	;**lda** = load	
001D	E10005	**sta** factor,d	;**sta** = store	
0020	C10005	floop: **lda** factor,d	;**lda** = load	
0023	800001	**suba** 0x0001,i	;**suba** = subtract	
0026	E10005	**sta** factor,d	;**sta** = store	
0029	800001	**suba** 0x0001,i	;**suba** = subtract	
002C	0A003E	**breq** prime	;**breq** = branch on equal to zero	
002F	C10003	**lda** number,d	;**lda** = load	
0032	810005	dloop: **suba** factor,d	;**suba** = subtract	
0035	0A0042	**breq** notprime	;**breq** = branch on equal to zero	(3)
0038	080020	**brlt** floop	;**brlt** = branch on less than zero	
003B	040032	**br** dloop	;**br** = branch	
003E	410007	prime: **stro** pr,d	;**stro** = output character string	
0041	00	**stop**		
0042	41000D	notprime:**stro** npr,d	;**stro** = output character string	
0045	00	**stop**		
		.end		

The memory location column will help you deduce which part of the instruction is in which memory cell. The bold numbers in brackets on the listing above are just to indicate an instruction referred to in the questions that follow.

Pretend to be a (Pep/8) compiler and run through the program. Include the program counter, instruction register, and other important components of the computer in your layout to help you answer the following questions. For more detailed elaboration of instruction meaning, please use the tables of descriptions of assembly code provided below.

(a) What will the **memory addresses 0003 and 0004** store right after the executing of the instruction at **(2)** with an input of 2?

0002

✔

(b) What will be the **output of the program** if 2 is input at instruction **(2)**?

prime

✔

(c) Will the program terminate if the input is 0? No ✔

(d) What will be the contents of the **program counter** after instruction **(0)** has been executed (give you answer in Base-16)?

0017

✔

(e) What addresses in memory will be used to store the value referred to by "factor" (give your answer in Base-16)?

0005

✔ and

0006

✔ .

(f) What is the meaning of the directive at **(1)**? Creates a binary string for a number ✔

(g) Suppose you are running through the program with input 7. Answer the following questions (1-9) with this input.

(1)What do the **memory addresses 0003 and 0004** store right after executing the instruction at **(2)**? Give your answer is hexadecimal.

0007

✔

(2) What number is stored in the **accumulator** after the instruction at **(3)** is executed for the first time. Give you answer in binary with the correct number of digits.

0000000000000001

✔ .

(3) What will be stored in **memory locations 0005 and 0006** after the instruction at **(3)** is executed **for the second time**. Give you answer in hexadecimal.

00006

✔

(4) What is stored in the **instruction register** immediately after the instruction at **(3)** is executed for the second time? (Give your answer in binary with the correct bit count)

00001000

✔

(5) What is stored in the **operand specifier** immediately after the instruction at **(3)** is executed for the second time? (Give your answer in binary with the correct bit count)

0000000000100000

✔

(6) What is stored in the **program counter** while the instruction at **(3)** is executed for the second time? (Give your answer in binary with the correct bit count)

0000000000111000

✔ .

(7) What is stored in the **Status bit N** immediately after the instruction at **(3)** is executed for the second time?

1

✔

(8) What is stored in the **Status bit Z** immediately after the instruction at **(3)** is executed for the second time?

0

✔

(9) What will be the **output of the program** once the program terminates?

prime

✔ .

(h) Now that you have studied the program, deduce what will be the **output of the program** if you input 100.

not prime

✔

You may use the following list of Assembly instructions and descriptions:

Mnemonic, mode	**Hex**	**Binary**	**Instruction**
			Load/store
LDA,i	C0	110b r000	Load into AC from immediate
LDA,d	C1	110b r001	Load into AC from memory address
STA,i	E0	1110 r000	[ILLEGAL INSTRUCTION]
STA,d	E1	111b r001	Store from AC into memory
	If "r" bit is 1, use X (index) register; else (accumulator) If "b" bit is 1, one bye only.		
			Arithmetic instructions
ADDA,i	70	0111 r000	Add AC + immediate (into AC)
ADDA,d	71	0111 0001	Add AC + memory (into AC)
SUBA,i	80	1000 r000	Subtract AC - immediate (into AC)
SUBA,d	81	1000 r001	Subtract AC - memory (into AC)
			Logical instructions
ANDA,i	90	1001 r000	AND AC + immediate (into AC)
ANDA,d	91	1001 r001	AND AC + memory (into AC)
ORA,i	A0	1010 r000	OR AC + immediate (into AC)
ORA,d	A1	1010 r001	OR AC + memory (into AC)
CPA,i	B0	1011 r000	Compare AC with immediate
CPA,d	B1	1011 r001	Compare AC with memory
NOTA	18	0001 100r	Invert every bit (1-s complement)
NEGA	1A	0001 101r	Negate the value (2-s complement)
ASLA	1C	0001 110r	Arithmetic shift left (with sign ext.)
ASRA	1E	0001 111r	Arithmetic shift right (end-off).
ROLA	20	0010 000r	Rotate left
RORA	22	0010 001r	Rotate right
			I/O instructions
CHARI,d	49	0100 1001	Character input to memory
CHARO,i	50	0100 1000	Character output from immediate
CHARO,d	51	0101 1001	Character output from memory
DECI,d	31	0011 0001	Read a decimal number into memory.
DECO,i	38	0011 1000	Write a decimal number from immediate.
DECO,d	39	0011 1001	Write a decimal number from memory.
STRO	40	0100 0000	String output trap
			Branch instructions
STOP	00	0000 0000	Stop the computer!

BR	04	0000 010x	Branch to specified address
BRLE	06	0000 011x	Branch if less-than-or-equal
BRLT	08	0000 100x	Branch if less-than
BREQ	0A	0000 101x	Branch if equal
BRNE	0C	0000 110x	Branch if not equal
BRGE	0E	0000 111x	Branch if greater-or-equal
BRGT	00	0000 000x	Branch if greater
BRV	02	0000 001x	Branch if overflow
BRC	04	0000 010x	Branch if carry
CALL	06	0000 011x	Call subroutine
RETn	58	0101 1nnn	Return from subroutine (with n bytes)
	If "x" bit is 1, use X (index) register; else immediate.		
			Stack and other instructions
ADDSP	60	0110 0000	Add to stack pointer
SUBSP	61	0110 0001	Subtract from stack pointer
MOVSPA	02	0000 0010	Move stack pointer to A
MOVFLGA	03	0000 0011	Move flags (NZVC) to A
RETTR	01	0000 0001	Return from trap
NOP	2n	0010 nnnn	Unary and non-unary no-op trap

Pseudo-op	Argument	Meaning of Instruction
.ASCII	"Str\x00"	Represents a string of ASCII bytes
.BLOCK	Number of bytes	Creates a block of bytes
.WORD	Value	Creates a word and stores a value in it
.END		Signals the end of the assembly language list

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Question **7**

Correct

Mark 1.00 out of 1.00

To which parts of the system are directives given that provide hints using predefined alphabetical strings (e.g., ".BLOCK")?

- a. assembler ✔
- b. processor and assembler
- c. memory
- d. processor

Your answer is correct.

The correct answer is:
assembler

Question **8**

Correct

Mark 1.00 out of 1.00

What is the meaning of the following machine language instruction in Pep-8? Note that the instruction specifier and operand specifier are separated by a space.

11000001 0000000000000111

You may use the following list of Pep-8 instructions:

Mnemonic	Operand Mode specifier	Meaning of Instruction
Stop		Stop execution
LDA	0x008B,i	Load 008B into Register A
LDA	0x008B,d	Load the contents of location 008B into Register A
STA	0x008B,d	Store the contents of Register A into location 008B
ADDA	0x008B,i	Add 008B into Register A
ADDA	0x008B,d	Add the contents of location 008B to Register A
SUBA	0x008B,i	Subtract 008B from Register A
SUBA	0x008B,d	Subtract the contents of location 008B from Register A
BR		Branch to the location specified in the operand specifier
CHARI	0x008B,d	Read a character and store it in location 008B
CHARO	0x008B,i	Write the character 8B
CHARO	0x008B,d	Write the character stored in location 008B
DECI	0x008B,d	Read a decimal number and store it in location 008B
DECO	0x008B,i	Write the decimal number 139 (8B in hex)
DECO	0x008B,d	Write the decimal number stored in location 008B

Pseudo-op	Argument	Meaning of Instruction
.ASCII	"Str\x00"	Represents a string of ASCII bytes
.BLOCK	Number of bytes	Creates a block of bytes
.WORD	Value	Creates a word and stores a value in it
.END		Signals the end of the assembly language list

Select one:

- a. Load 7 into the A register
- b. Subtract the value 7 from the value in the A register
- c. Subtract the value stored in memory at position 7 from the value in the A register
- d. Load the value stored in memory at position 7 into the A register ✔

Your answer is correct.

The correct answer is: Load the value stored in memory at position 7 into the A register

◄ Slides Low Level Programming (COMPLETE)

Jump to...

Assembly Code (Multiply A x B) ►