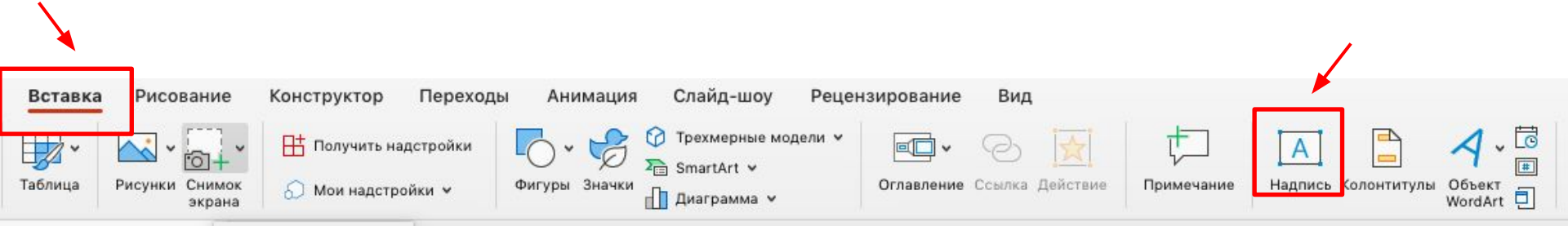
Workbook

Name _____ grade_____

Instruction for implementation
- Insert the correct answer using the function insert -> inscription



Assembly language instructions

Instruction		Explanation
Op code	Operand	
LDD	<address>	Direct addressing. Load the contents of the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
CMP	<address>	Compare contents of ACC with contents of <address>
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to the screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

Task 1

(b) Trace the assembly language program using the trace table.

```
300   LDD   321
301   INC
302   STO   323
303   LDI   307
304   INC
305   STO   322
306   END
307   320
320   49
321   36
322   0
323   0
```

Trace table:

Accumulator	Memory address			
	320	321	322	323
	49	36	0	0

[6]

Task 2

8 The table shows assembly language instructions for a processor which has one general purpose register, the Accumulator (ACC) and an Index Register (IX).

Instruction		Explanation
Op code	**Operand**	
LDD	<address>	Direct addressing. Load the contents of the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
CMP	<address>	Compare contents of ACC with contents of <address>
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to the screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

The diagram shows the contents of the main memory:

Main memory

800	0110 0100
801	0111 1100
802	1001 0111
803	0111 0011
804	1001 0000
805	0011 1111
806	0000 1110
807	1110 1000
808	1000 1110
809	1100 0010
⋮	
2000	1011 0101

(a) (i) Show the contents of the Accumulator after execution of the instruction:

LDD 802

Accumulator: | | | | | | | | |

[1]

(ii) Show the contents of the Accumulator after execution of the instruction:

LDX 800

Index Register:	0	0	0	0	1	0	0	1
Accumulator:								

Explain how you arrived at your answer.

.....

.....

.....

.....[3]

Task 3

Instruction		Explanation
Op code	Operand	
LDD	<address>	Direct addressing. Load the contents of the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
CMP	<address>	Compare contents of ACC with contents of <address>
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to the screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

(b) (i) Complete the trace table below for the following assembly language program. This program contains denary values.

100	LDD 800
101	ADD 801
102	STO 802
103	LDD 803
104	CMP 802
105	JPE 107
106	JPN 110
107	STO 802
108	OUT
109	JMP 112
110	LDD 801
111	OUT
112	END
⋮	
800	40
801	50
802	0
803	90

Selected values from the ASCII character set:

ASCII code	40	50	80	90	100
Character	(	2	P	Z	d

Trace table:

ACC	Memory address				OUTPUT
	800	801	802	803	
	40	50	0	90	

[4]

Task 4.1

9 The table shows assembly language instructions for a processor which has one general purpose register, the Accumulator (ACC) and an index register (IX).

Instruction		Explanation
Op code	**Operand**	
LDD	<address>	Direct addressing. Load the contents of the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
INC	<register>	Add 1 to the contents of the register (ACC or IX).
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX).
CMP	<address>	Compare contents of ACC with contents of <address>.
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

(a) The diagram shows the current contents of a section of main memory and the index register:

60	0011 0010
61	0101 1101
62	0000 0100
63	1111 1001
64	0101 0101
65	1101 1111
66	0000 1101
67	0100 1101
68	0100 0101
69	0100 0011
...	
1000	0110 1001

Index register: 0 0 0 0 1 0 0 0

(i) Show the contents of the Accumulator after the execution of the instruction:

LDX 60

Accumulator: [ ][ ][ ][ ][ ][ ][ ][ ]

Show how you obtained your answer.

.....

.....

.....

.....[2]

(ii) Show the contents of the index register after the execution of the instruction:

DEC IX

Index register: [ ][ ][ ][ ][ ][ ][ ][ ]

[1]

Task 4.2

(b) Complete the trace table on the opposite page for the following assembly language program.

50	LDD 100
51	ADD 102
52	STO 103
53	LDX 100
54	ADD 100
55	CMP 101
56	JPE 58
57	JPN 59
58	OUT
59	INC IX
60	LDX 98
61	ADD 101
62	OUT
63	END
...	
100	20
101	100
102	1
103	0

IX (Index Register)	1

Selected values from the ASCII character set:

ASCII Code	118	119	120	121	122	123	124	125
Character	v	w	x	y	z	{	|	}

Trace table:

Instruction address	Working space	ACC	Memory address				IX	OUTPUT
			100	101	102	103		
			20	100	1	0	1	
50								
51								
52								
53								
54								
55								

[7]

Task 5

4 The following table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC) and an Index Register (IX).

Instruction			Explanation
Op code (mnemonic)	Operand	Op code (binary)	
LDM	#n	0000 0001	Immediate addressing. Load the denary number n to ACC.
LDD	<address>	0000 0010	Direct addressing. Load the contents of the location at the given address to ACC.
LDI	<address>	0000 0101	Indirect addressing. At the given address is the address to be used. Load the contents of this second address to ACC.
LDX	<address>	0000 0110	Indexed addressing. Form the address from <address> + the contents of the Index Register (IX). Copy the contents of this calculated address to ACC.
LDR	#n	0000 0111	Immediate addressing. Load number n to IX.
STO	<address>	0000 1111	Store the contents of ACC at the given address.

The following diagram shows the contents of a section of main memory and the Index Register (IX).

(a) Show the contents of the Accumulator (ACC) after each instruction is executed.

IX

0	0	0	0	0	0	1	1

Address	Main Memory contents
495	13
496	86
497	92
498	486
499	489
500	496
501	497
502	499
503	502

(i) LDM #500

ACC .....[1]

(ii) LDD 500

ACC .....[1]

(iii) LDX 500

ACC .....[1]

(iv) LDI 500

ACC .....[1]

(b) Each machine code instruction is encoded as 16-bits (8-bit op code followed by an 8-bit operand).

Write the machine code for the following instructions:

LDM #17

LDX #97

[3]

Task 6

5 The following table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC), and an Index Register (IX).

Instruction			Explanation
Op code (mnemonic)	Operand	Op code (binary)	
LDD	<address>	0001 0011	Direct addressing. Load the contents of the location at the given address to the Accumulator (ACC).
LDI	<address>	0001 0100	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC.
LDX	<address>	0001 0101	Indexed addressing. Form the address from <address> + the contents of the Index Register. Copy the contents of this calculated address to ACC.
LDM	#n	0001 0010	Immediate addressing. Load the denary number n to ACC.
LDR	#n	0001 0110	Immediate addressing. Load denary number n to the Index Register (IX).
STO	<address>	0000 0111	Store the contents of ACC at the given address.

The following diagram shows the contents of a section of main memory and the Index Register (IX).

(a) Show the contents of the Accumulator (ACC) after each instruction is executed.

IX	0	0	0	0	0	1	1	0

(i) LDD 355

ACC ..... [1]

(ii) LDM #355

ACC ..... [1]

(iii) LDX 351

ACC ..... [1]

(iv) LDI 355

ACC ..... [1]

Address	Main memory contents
350	
351	86
352	
353	
354	
355	351
356	
357	22
358	

(b) Each machine code instruction is encoded as 16 bits (8-bit op code followed by an 8-bit operand).

Write the machine code for these instructions:

LDM #67

LDX #7

[3]

Task 7

4 The following table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC) and an Index Register (IX).

Instruction			Explanation
Op code (mnemonic)	Operand	Op code (binary)	
LDM	#n	0000 0001	Immediate addressing. Load the denary number n to ACC.
LDD	<address>	0000 0010	Direct addressing. Load the contents of the location at the given address to ACC.
LDI	<address>	0000 0101	Indirect addressing. At the given address is the address to be used. Load the contents of this second address to ACC.
LDX	<address>	0000 0110	Indexed addressing. Form the address from <address> + the contents of the Index Register (IX). Copy the contents of this calculated address to ACC.
LDR	#n	0000 0111	Immediate addressing. Load number n to IX.
STO	<address>	0000 1111	Store the contents of ACC at the given address.

The following diagram shows the contents of a section of main memory and the Index Register (IX).

(a) Show the contents of the Accumulator (ACC) after each instruction is executed.

IX

0	0	0	0	0	0	1	1

Address	Main Memory contents
495	13
496	86
497	92
498	486
499	489
500	496
501	497
502	499
503	502

(i) LDM #500

ACC .....[1]

(ii) LDD 500

ACC .....[1]

(iii) LDX 500

ACC .....[1]

(iv) LDI 500

ACC .....[1]

(b) Each machine code instruction is encoded as 16-bits (8-bit op code followed by an 8-bit operand).

Write the machine code for the following instructions:

LDM #17

LDX #97

[3]

Task 8

4 The table shows assembly language instructions for a processor which has one general purpose register, the Accumulator (ACC) and an index register (IX).

Instruction		Explanation
Op code	Operand	
LDD	<address>	Direct addressing. Load the contents of the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
INC	<register>	Add 1 to the contents of the register (ACC or IX).
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX).
CMP	<address>	Compare contents of ACC with contents of <address>.
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

The diagram shows the contents of the index register:

Index register:

1	1	0	0	1	1	0	1

(a) Show the contents of the index register after the execution of the instruction:

```
INC IX
```

Index register:

[1]

(b) Complete the trace table on the opposite page for the following assembly language program.

20	LDX 90
21	DEC ACC
22	STO 90
23	INC IX
24	LDX 90
25	DEC ACC
26	CMP 90
27	JPE 29
28	JPN 31
29	ADD 90
30	OUT
31	ADD 93
32	STO 93
33	OUT
34	END
⋮	
90	2
91	90
92	55
93	34

IX	2

Trace table:

Instruction	Working space	ACC	Memory address				IX	OUTPUT
			90	91	92	93		
			2	90	55	34	2	
20								
21								
22								
23								
24								
25								
26								

[7]

cted values from the ASCII character set:

ll Code	65	66	67	68	69	70	71	72
Character	A	B	C	D	E	F	G	H

Task 9

Part of the assembly language code for updating LOWREG is:

Label	Op code	Operand
LOWTEMP:		15
LOWREG:		B00000000
COUNTER:		1
START:	LDR	#0
LOOP:	LDX	8000
	CMP	LOWTEMP
	JGE	TEMPOK
	LDD	LOWREG
	OR	COUNTER
	STO	LOWREG
TEMPOK:	LDD	COUNTER
Q1:	CMP	#32
	JPE	HEATON
	ADD	COUNTER
	STO	COUNTER
	INC	IX
	JMP	LOOP
HEATON:	LDD	LOWREG

(i) The code uses six memory locations to store the temperature readings. It stores readings for sensors 1 to 6 at addresses 8000 to 8005.

At a particular time, the memory locations store the following data.

8000	8001	8002	8003	8004	8005
17	14	15	15	16	14

Dry run the assembly language code starting at START and finishing when the loop has been processed twice.

LOWTEMP	LOWREG	COUNTER	ACC	IX
15	B00000000	1		

[4]

Task 10

The intruder system is set up so that the alarm will only sound if two or more sensors have been triggered.
An assembly language program has been written to process the contents of the memory location.

The table shows part of the instruction set for the processor used.

Instruction		Explanation
Op code	Operand	
LDD	<address>	Direct addressing. Load the contents of the given address to ACC
STO	<address>	Store the contents of ACC at the given address
INC	<register>	Add 1 to the contents of the register (ACC or IX)
ADD	<address>	Add the contents of the given address to the contents of ACC
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
CMP	#n	Compare the contents of ACC with the number n
JMP	<address>	Jump to the given address
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True
JGT	<address>	Following a compare instruction, jump to <address> if the content of ACC is greater than the number used in the compare instruction
END		End the program and return to the operating system

(c) Part of the assembly code is:

	Op code	Operand
SENSORS:		B00001010
COUNT:		0
VALUE:		1
LOOP:	LDD	SENSORS
	AND	VALUE
	CMP	#0
	JPE	ZERO
	LDD	COUNT
	INC	ACC
	STO	COUNT
ZERO:	LDD	VALUE
	CMP	#8
	JPE	EXIT
	ADD	VALUE
	STO	VALUE
	JMP	LOOP
EXIT:	LDD	COUNT
TEST:	CMP	...
	JGT	ALARM

(i) Dry run the assembly language code. Start at LOOP and finish when EXIT is reached.

BITREG	COUNT	VALUE	ACC
B00001010	0	1	

[4]