8 The following table shows part of the instruction set for a processor. The processor has two registers, the Accumulator (ACC) and the Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second 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
LDR	#n	Immediate addressing. Load the number n to IX
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
SUB	<address>	Subtract the contents of the given address from the 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)
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001010 & denotes a hexadecimal number, e.g. &4A		

(a) The current contents of memory are shown:

Address	Data
19	25
20	23
21	2
22	4
23	15
24	50
25	22

The current contents of the ACC and IX are shown:

ACC	50
IX	20

Complete the table by writing the content of the ACC and the IX after each set of instructions has run.

	Instructions	ACC content	IX content
1	LDM #19 DEC ACC		
2	LDD 23 ADD 19		
3	LDI 25 INC ACC		
4	LDR #21 LDX 2		

[5]

(b) The instruction set also includes these bit manipulation instructions:

Instruction		Explanation
Opcode	**Operand**	
AND	#n/Bn/&n	Bitwise AND operation of the contents of ACC with the operand
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#n/Bn/&n	Bitwise XOR operation of the contents of ACC with the operand
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#n/Bn/&n	Bitwise OR operation of the contents of ACC with the operand
OR	<address>	Bitwise OR operation of the contents of ACC with the contents of <address>
LSL	#n	Bits in ACC are shifted logically n places to the left. Zeros are introduced on the right-hand end
LSR	#n	Bits in ACC are shifted logically n places to the right. Zeros are introduced on the left-hand end.
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001010 & denotes a hexadecimal number, e.g. &4A		

The current content of the ACC is shown:

ACC	1	0	0	1	1	0	1	0

(i) The table has three sets of instructions. The binary number 10011010 is reloaded into the ACC before each set of instructions is run.

Complete the table by writing the content of the ACC after each set of instructions has run.

	Instructions	ACC content
1	LSL #2	
2	ADD #5 AND #30	
3	OR B11110010 INC ACC	

[3]

(ii) Explain how bit manipulation can be used to test whether the binary number stored in the ACC represents an odd denary number.

Write the bit manipulation instruction that will be used.

Explanation .....

.....

.....

.....

.....

Instruction .....

[3]