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

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
STO	<address>	Store the contents of ACC at the given address
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX)
JMP	<address>	Jump to the given address
CMP	<address>	Compare the contents of ACC with the contents of <address>
CMP	#n	Compare the contents of ACC with number n
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True
END		Return control to the operating system

ACC denotes Accumulator
<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:

Address	Instruction
50	47
51	48
52	49
53	50
...	
500	LDM #50
501	DEC ACC
502	CMP 51
503	JPE 505
504	JMP 501
505	STO 50
506	SUB #10
507	STO 51
508	END

(i) Trace the program currently in memory using the following trace table.

Instruction address	ACC	Memory address			
		50	**51**	**52**	**53**
		47	48	49	50

[3]

(ii) Complete the table by identifying **and** describing **two** modes of addressing that are **not** used in the program in part **(a)**.

Mode of addressing	Description
....	....
....	....

[4]

(b) The table shows part of the instruction set for a processor. The processor has one register: the Accumulator (ACC).

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>

<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

Write the bit manipulation instructions that can be used to set only the most significant bit to 1 in an 8-bit register.

The instructions need to work on a register that contains any 8-bit binary number.

.....

..... [2]