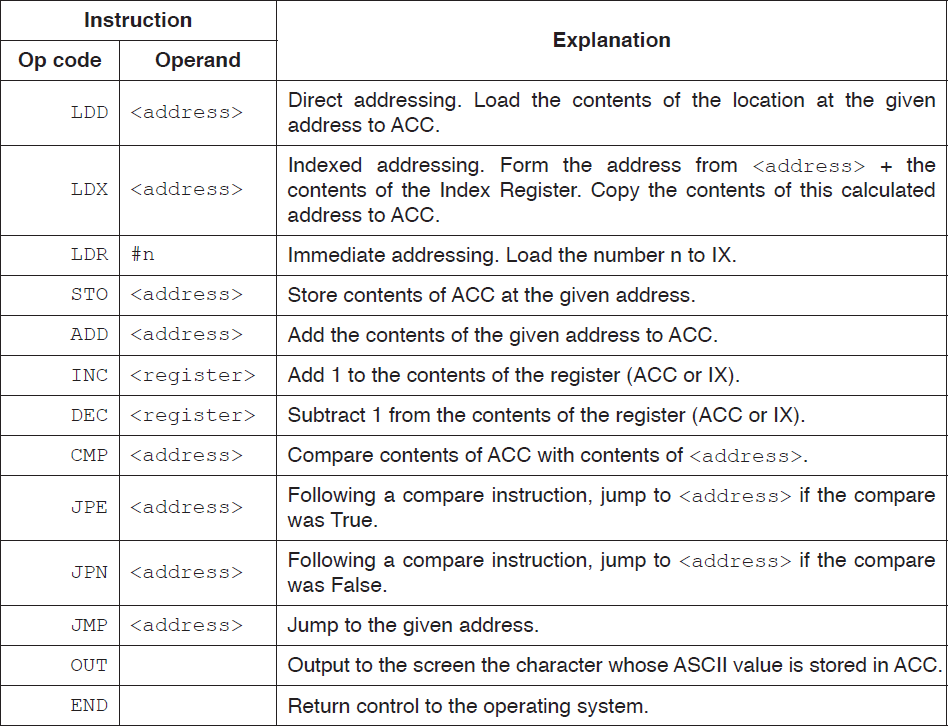
**6.1 Assembly Language**

**Name \_\_\_\_\_\_\_\_\_\_\_ class\_\_\_\_\_\_\_\_\_\_\_\_\_**

1

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



**(a) (i)** State what is meant by **direct addressing** and **indirect addressing**.

Direct addressing

..........................................................................................................

............................................................................................................[1]

Indirect addressing .

...........................................................................................................

........................................................................................................... [1]

**(ii)** Explain how the instruction ADD 20 can be interpreted as either direct or indirect

addressing.

Direct addressing

...........................................................................................................

............................................................................................................. [1]

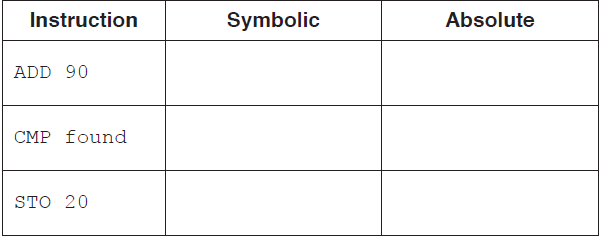
Indirect addressing

............................................................................................................

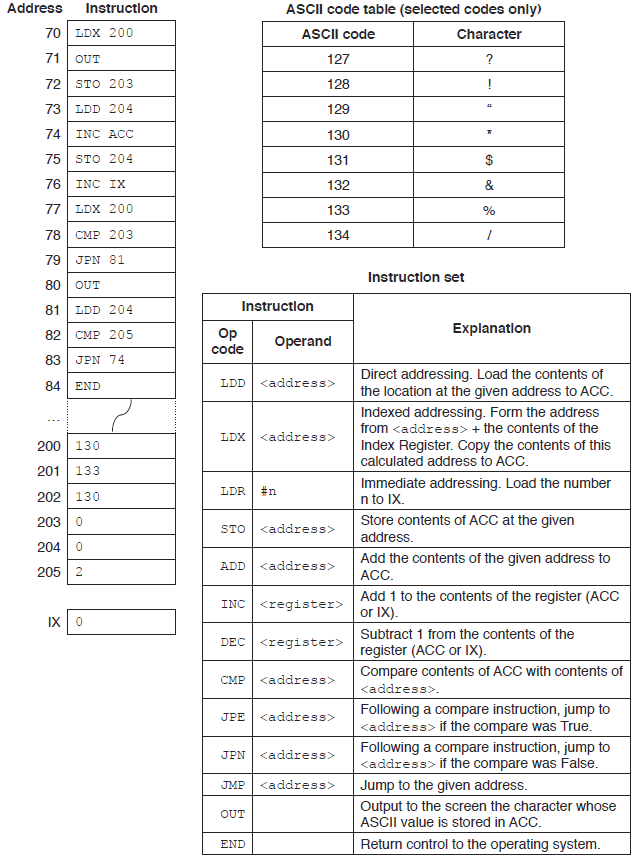
............................................................................................................... [1]

The assembly language instructions in the following table use either symbolic addressing or absolute addressing.

Tick (✔) **one** box in each row to indicate whether the instruction uses symbolic or absolute addressing.

[2]

**(d)** The current contents of the main memory, Index Register (IX) and selected values from the ASCII character set are provided with a copy of the instruction set.



Complete the trace table for the given assembly language program. Fill in the blanks only when the content changes.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction**  **address** | ACC | **Memory address** | | | | | | IX | OUTPUT |
| 200 | 201 | 202 | 203 | 204 | 205 |
| 70 | 130 | 130 | 133 | 130 | 0 | 0 | 2 | 0 |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

[8]

2.

A hardware device to affect the temperature of each tank is on or off depending on the value of a bit in memory location 6753. If bit 4 is 1, then the hardware device in fish tank 4 is on.

Write **assembly language** instructions to set bit 4 of memory location 6753 to 1 without changing any other bits. Use the instruction set provided.

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

..................................................................................................................................[3]

**Instruction set**

LDD <address> Direct addressing. Load the contents of the location at the given

address to ACC.

STO <address> Store the contents of ACC at the given address.

AND #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 Bitwise XOR operation of the contents of ACC with the operand.

OR #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 address or a symbolic address.

3.

An assembly language program can contain both **macros** and **directives**.

**(i)** Explain what is meant by these terms.

Macro ............................................................................................................................

.......................................................................................................................................

...................................................................................................................................[1]

Directive ........................................................................................................................

.......................................................................................................................................

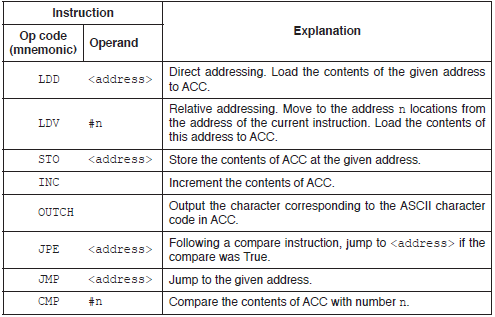
...................................................................................................................................[1]

**(ii)** Give an example of the use of a directive.

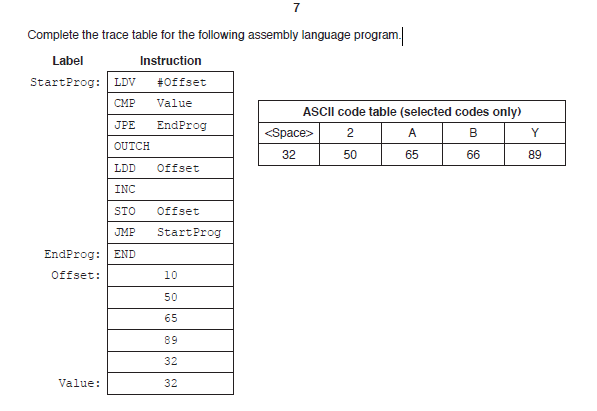
.......................................................................................................................................

...................................................................................................................................[1]

**(d)** 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).



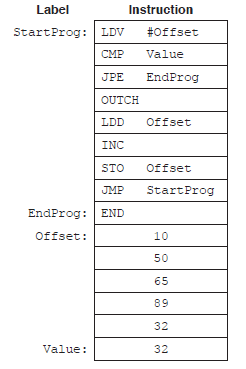
Complete the trace table for the following assembly language program.



|  |  |  |
| --- | --- | --- |
| ACC | Offset | OUTPUT |
|  | 10 |  |
| 50 |  | 2 |
| 10 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[5]

**(e)** The program given in **part (d)** is to be translated using a two-pass assembler. The program has been copied here for you.

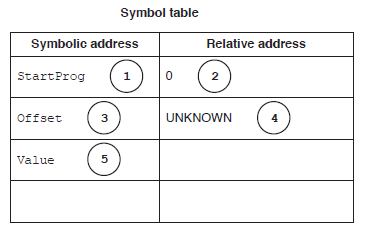


On the first pass, the assembly process adds entries to a symbol table.

The following symbol table shows the first five entries, part way through the first pass.

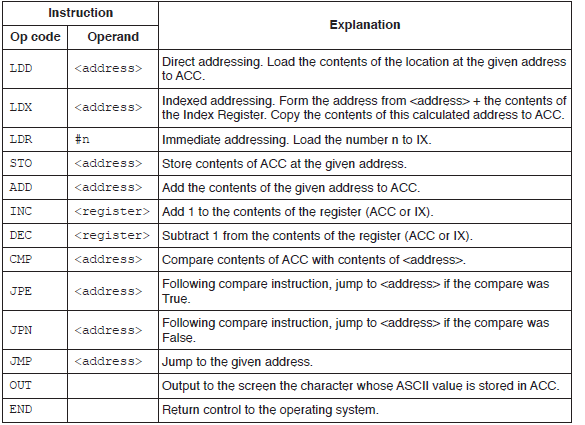
The circular labels show the order in which the assembler made the entries to the symbol table.

Complete the symbol table. Use circular labels to show the order in which the assembler makes the entries.

[6]

4.

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



**(a)** State what is meant by **relative addressing** and **indexed addressing**.

Relative addressing

.....................................................................................................................

.......................................................................................................................................

Indexed addressing

.......................................................................................................................................

.......................................................................................................................................

[2]

**(b)** The current contents of a general purpose register (X) are:



**(i)** The contents of X represent an unsigned binary integer.

Convert the value in X into denary.

.......................................................................................................................................[1]

**(ii)** The contents of X represent an unsigned binary integer.

Convert the value in X into hexadecimal.

.......................................................................................................................................[1]

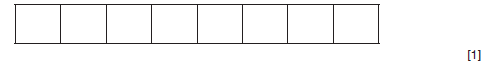
**(iii)** The contents of X represent a two’s complement binary integer.

Convert the value in X into denary.

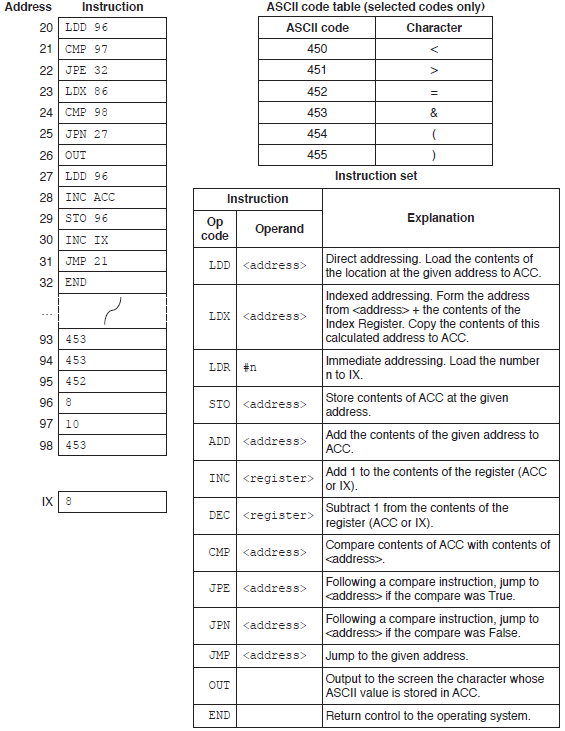
.......................................................................................................................................[1]

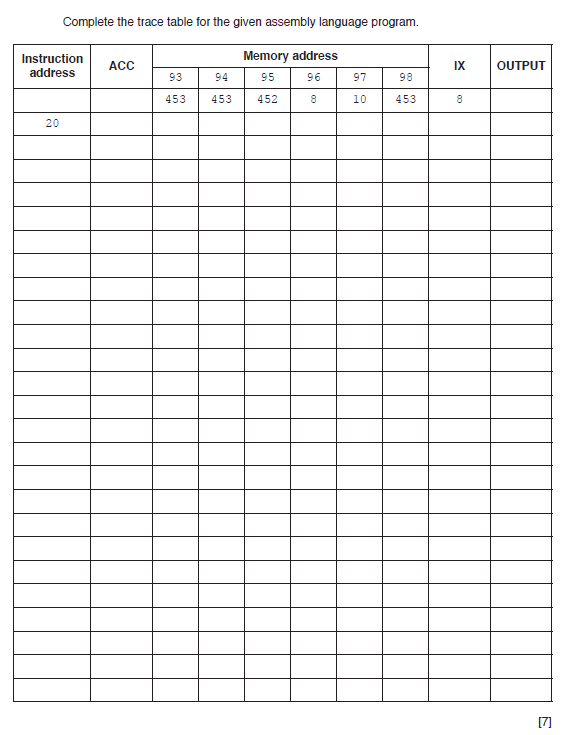
**(iv)** Show the result on the general purpose register (X) after the following instruction is run.

INC X

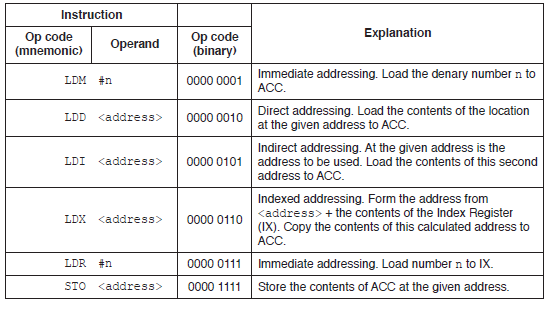


**(c)** The current contents of the main memory, Index Register (IX) and selected values from the ASCII character set are provided with a copy of the instruction set.



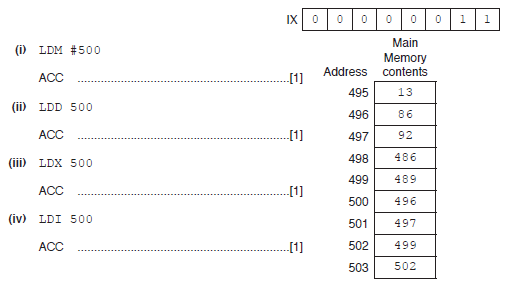


5. The following table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC) and an Index Register (IX).



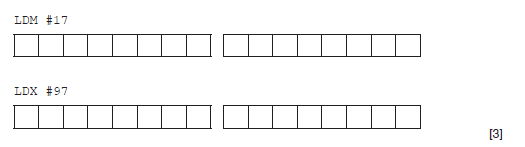
The following diagram shows the contents of a section of main memory and the Index Register (IX).

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



**（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:



**(c)** Using an 8-bit operand, state the maximum number of memory locations, in denary, that can be directly addressed.

................................................................................................................................. [1]

**(d)** Computer scientists often write binary representations in hexadecimal.

**(i)** Write the hexadecimal representation for this instruction:

.

...............................................................................................................................[2]

**(ii)** A second instruction has been written in hexadecimal as:

**05 3F**

Write the equivalent assembly language instruction, with the operand in denary.

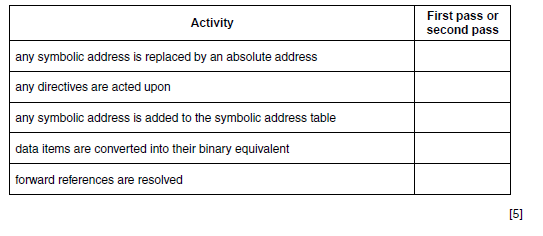
.........................................................................................................................[2]

6.

Assemblers translate from assembly language to machine code. Some assemblers scan the assembly language program twice; these are referred to as two-pass assemblers.

The following table shows five activities performed by two-pass assemblers.

Write 1 **or** 2 to indicate whether the activity is carried out during the first pass or during the second pass..



7.

**(i)** The accumulator is loaded with the value of byte 1 from location 106.

Write the assembly language instruction to check whether the reading in byte 2 came

from location 4.

LDD 106 // data loaded from address 106

.......................................................................................................................................[4]

**(ii)** Write the assembly language instruction to set the flag (bit 0) of the byte contained in the

accumulator to 1.

..................................................................................................................................[2]