# Worksheet 3 Programming language classification

**Task 1**

The following table shows some of the instructions available in an imaginary computer which uses 4 bits for the opcode and 4 bits for the operand.

|  |  |
| --- | --- |
| **Instruction** | **Meaning** |
| 0000 | Load the value stored in memory location specified by the operand into the accumulator |
| 0001 | Store the value in the accumulator in memory location specified by the operand |
| 0010 | Add the value specified in the operand to the value in the accumulator |
| 0011 | Compare the contents of the accumulator with the contents of the location specified by the operand |
| 0100 | Jump to the address held in the operand if the accumulator held the lesser value in the last comparison |
| 0101 | Jump to the address held in the operand if the accumulator held the greater value in the last comparison |
| 0110 | Jump to the address held in the operand |
| 1000 | Stop |

1. (a) Add comments to each of the following instructions to say what it does. [9]

memory instruction

address

1 0000 1001 ;load value from address 9 into accumulator

2 0011 1010

3 0100 0110

4 0001 1011

5 0110 1000

6 0000 1010

7 0001 1011

8 1000 0000

9 0000 0110

10 0000 0111

(b) What will be stored in memory address 11 if memory address 9 holds 6 and memory address 10 holds 12? [1]

(c) What is the purpose of the program, in general terms? [2]

2. Write a machine code program to add the values in memory addresses 12, 13 and 14. Store the result in location 15. [4]

**Task 2**

Some assembly code instructions for a processor are given in the table below.

|  |  |
| --- | --- |
| **Instruction** | **Meaning** |
| LDA | Load the value stored in memory location specified by the operand into the accumulator |
| STO | Store the value in the accumulator in memory location specified by the operand |
| ADD | Add the value specified in the operand to the value in the accumulator |
| CMP | Compare the contents of the accumulator with the contents of the location specified by the operand |
| BLT | Jump to the address held in the operand if the accumulator held the lesser value in the last comparison |
| BGT | Jump to the address held in the operand if the accumulator held the greater value in the last comparison |
| JMP | Jump to the address held in the operand |
| STOP | Stop |

1. Rewrite the program in Task 1, Question 1 in Assembly language. [10]

2. Write a pseudocode algorithm that will perform the same task as the machine code program in Task 1 Question 1. [4]

3. What are the **disadvantages** of high-level programming languages compared with low-level languages? [2]