

Languages and translators

Q1 Some low-level programming languages use mnemonics.

(i) State the purpose of a mnemonic.

(1)

.....
.....

(ii) State the type of low-level programming language that uses mnemonics.

(1)

.....

(iii) Mnemonics are one characteristic of some low-level languages.

Describe **one other** characteristic of a low-level language.

(2)

.....
.....

(iv) State the name of the high-level programming language translator that executes a line of code immediately after translating it.

(1)

.....

Q2. Define the term 'translation'.

(1)

.....
.....

Q3. Describe **two** ways a compiler differs from an interpreter.

(4)

1

.....

.....

.....

2

.....

.....

.....

Q4. When using a compiler to translate source code to machine code, the translation only needs to be done once.

Give **three other** features of a compiler.

(3)

1

.....

2

.....

3

.....

ANSWERS

1. (i) A **mnemonic** acts as a memory aid. It also makes code more **readable**. They make it easier for programmers to understand and remember the commands.

(ii) Assembly language

(iii) Low-level languages provide **little or no abstraction** from the underlying **hardware architecture** of the computer.

Programmers can directly manipulate **registers, memory**, and other hardware elements. This is why it is used in embedded systems and device drivers.

One instruction translates to one machine code instruction.

Code executes faster

Code works on one type of processor only.

Code is memory efficient.

(iv) Interpreter

2. Translation is converting a high level language, or low level assembly language, into binary machine code.
- 3 and 4 **Compiler**: Translates the entire source code (written in a high-level language) into machine code before execution. The resulting machine code is a standalone program that can run directly on the computer without the compiler. Compiled programs generally run faster because the machine code is optimized for the target system and doesn't require translation during execution. Compilers produce an error report after the entire code is executed. Compilers hide the source code from the end user. This protects the source code from theft. But it does mean that for a developer to fix any errors, they have to go back to the source code, fix the error and then re-compile it.
Interpreter: Executes the source code line by line when the program is run. It reads each line, translates it to machine code instructions, and executes them immediately. Interpreted programs can be slower because the translation to machine code happens at runtime for each line of code. Interpreters stop when they first find an error. This makes finding and fixing errors more quickly. The source code is visible to the user which is useful for the developer. But users would be able to see the code and would need an interpreter installed in order to run the code.