Briefly (and, necessarily, without being too precise) explain the difference between an algorithm and a program. Suppose we have an algorithm and wish to implement it in Python. How many different implementations are there of this algorithm? [5]

Give 3 general properties any programming language should have. [3]

Explain very briefly how a Prolog program computes. [4]

Give an advantage of compilation over interpretation, and of interpretation over compilation. [2]

In compilation, over 50% of the time taken can be spent on lexical analysis; that is, character handling. In moving from a program as a string of symbols to a token stream, which algebraic construction is usually used to define tokens? [2]

Give a regular expression that denotes the set of strings over the alphabet {a, b, c} consisting of strings with the property that there is always at most one c. [5]

Give a regular expression that denotes the set of strings over the alphabet {a, b} consisting of strings with the property that any a must always be followed by bb. [7]

Define carefully a finite state machine and explain how one is used to accept a set of strings. [8]

What is the fundamental principle of the research area known as ubiquitous computing? [2]

Give two illustrations of principles of Computational Thinking in the context of software. [2]

- Be easy to use, with its programs easy to read, write and understand
- Support abstraction so that adding new features and concepts should be possible
- Support testing, debugging and program verification
- Be inexpensive to use, in terms of execution time, memory usage and maintenance costs

An algorithm is a sequence of precise instructions that can be applied to specific data items. A program is the implantation of the algorithm in a form that can be executed by a computer, or at least compiled to a form that can be executed by a computer. There are many different implementations of an algorithm as a program.

- Compiled programs have faster execution.
- When a program is compiled it can be optimised at the same time, this can't be done through interpretation.
- Interpreted programs use memory better as only a few instructions need to be stored at once.

A Prolog program consists of a list of facts (atoms) and rules that can be applied to the facts. It then takes queries about the facts which it can answer using the atoms and the rules.

$$((a|b)*|c(a|b)*)|(a|b)*$$

Regular Expressions

$$M = (\Sigma, Q, \delta : Q \times \Sigma \to Q, q_0 \in Q, F)$$

Where

- Σ is some finite alphabet
- Q is some finite set of **states** with **initial state** q_0 and a set of **final states** F $\subseteq Q$
- $\delta : Q \times \Sigma \to Q$ is the **transition function**

$$(b|abb)*$$

Green Computing:An area of computer science involving energy conservation within the world of information technology. This involves writing the main unit of resource is energy expended.

Parallel processing: Certain kinds of problems can be conveniently represented as multiple communicating threads which help to structure code in a more modular manner, e.g., by modelling user interface components as separate threads.

The integration of computers and software into everyday objects and activities so that we can control remote aspects of our lives, mostly through RFID.

What is the relationship between regular expressions and finite state machines? [2]

A set of strings is represented by a regular expression if, and only if, it is accepted by a finite state machine