Programming languages are essential for software development. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Scripting and breakpointing is also part of this process. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Programs were mostly entered using punched cards or paper tape. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Computer programmers are those who write computer software. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Integrated development environments (IDEs) aim to integrate all such help. There exist a lot of different approaches for each of those tasks.