It is usually easier to code in "high-level" languages than in "low-level" ones. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Integrated development environments (IDEs) aim to integrate all such help. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. It is usually easier to code in "high-level" languages than in "low-level" ones. Programming languages are essential for software development. It is very difficult to determine what are the most popular modern programming languages. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Techniques like Code refactoring can enhance readability. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Whatever the approach to development may be, the final program must satisfy some fundamental properties.