Normally the first step in debugging is to attempt to reproduce the problem. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. 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. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. It is very difficult to determine what are the most popular modern programming languages. Unreadable code often leads to bugs, inefficiencies, and duplicated code. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. Programming languages are essential for software development. It is usually easier to code in "high-level" languages than in "low-level" ones. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Following a consistent programming style often helps readability. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.