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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Code-breaking algorithms have also existed for centuries. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Use of a static code analysis tool can help detect some possible problems. Also, specific user environment and usage history can make it difficult to reproduce the problem. However, readability is more than just programming style. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Also, specific user environment and usage history can make it difficult to reproduce the problem. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Use of a static code analysis tool can help detect some possible problems. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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.
Code-breaking algorithms have also existed for centuries. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.