Programs were mostly entered using punched cards or paper tape. Code-breaking algorithms have also existed for centuries. It is usually easier to code in "high-level" languages than in "low-level" ones. Code-breaking algorithms have also existed for centuries. 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. Code-breaking algorithms have also existed for centuries. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Use of a static code analysis tool can help detect some possible problems. Computer programmers are those who write computer software. 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. Many applications use a mix of several languages in their construction and use. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Following a consistent programming style often helps readability. Whatever the approach to development may be, the final program must satisfy some fundamental properties. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. It affects the aspects of quality above, including portability, usability and most importantly maintainability. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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.