Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Following a consistent programming style often helps readability. 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. 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. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Many applications use a mix of several languages in their construction and use. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Programming languages are essential for software development. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. Programmable devices have existed for centuries. Code-breaking algorithms have also existed for centuries. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Techniques like Code refactoring can enhance readability. Use of a static code analysis tool can help detect some possible problems. Scripting and breakpointing is also part of this process. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.