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. Programmable devices have existed for centuries. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Programs were mostly entered using punched cards or paper tape. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. 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. Techniques like Code refactoring can enhance readability. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Integrated development environments (IDEs) aim to integrate all such help. 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. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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.