This can be a non-trivial task, for example as with parallel processes or some unusual software bugs..  
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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Scripting and breakpointing is also part of this process.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
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.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
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.