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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
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.  
 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.  
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.  
There exist a lot of different approaches for each of those tasks.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 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.  
Use of a static code analysis tool can help detect some possible problems.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.