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..  
Normally the first step in debugging is to attempt to reproduce the problem.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
Use of a static code analysis tool can help detect some possible problems.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 Programs were mostly entered using punched cards or paper tape.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
Scripting and breakpointing is also part of this process.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Techniques like Code refactoring can enhance readability.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.