The ten rules are:[[1]](https://en.wikipedia.org/wiki/The_Power_of_10:_Rules_for_Developing_Safety-Critical_Code#cite_note-PowerOf10-1)

1. Avoid complex flow constructs, such as [goto](https://en.wikipedia.org/wiki/Goto) and [recursion](https://en.wikipedia.org/wiki/Recursion_(computer_science)).
2. All loops must have fixed bounds. This prevents runaway code.
3. Avoid [heap memory allocation](https://en.wikipedia.org/wiki/Memory_management#DYNAMIC) after [initialization](https://en.wikipedia.org/wiki/Initialization_(programming)).
4. Restrict functions to a single printed page.
5. Use a minimum of two [runtime assertions](https://en.wikipedia.org/wiki/Assertion_(software_development)#Assertions_for_run-time_checking) per function.
6. Restrict the scope of data to the smallest possible.
7. Check the return value of all non-void functions, or cast to void to indicate the return value is useless.
8. Use the [preprocessor](https://en.wikipedia.org/wiki/Preprocessor) only for [header files](https://en.wikipedia.org/wiki/Include_directive) and simple [macros](https://en.wikipedia.org/wiki/Macro_(computer_science)#Text-substitution_macros).
9. Limit pointer use to a single [dereference](https://en.wikipedia.org/wiki/Dereference_operator), and do not use [function pointers](https://en.wikipedia.org/wiki/Function_pointer).
10. Compile with all possible warnings active; all warnings should then be addressed before release of the software.

Jezeli kroki programuwymakaja dostepu strzezonego nalezy na samym starcie programu upewniac sie ze dostep jest zwalniany(plik, komunikacja etc)