1. **Do you agree or disagree with the third bullet point that “The tools we use have a profound (and devious!) influence on our thinking habits, and, therefore, on our thinking abilities”? Explain.**
   1. I absolutely agree with this. Therefore, in math classes they teach you how to find the answer using arithmetic and they show the theory and reasoning behind a formula before they show you how to solve the equation in a simpler way. Because if they just show you the simple way then you will never understand why it is that way and it will limit your thinking ability in the future with different problems. A lot of the times we will get stuck on a problem because we are trying to force the solution from the tools, we have instead of finding or creating new tools.
2. **What qualities did Dijkstra believe were essential for someone to become a competent programmer? Why do you believe these may be necessary or not?**
   1. Dijkstra said that a competent programmer has not only a mathematical inclination, but also an exceptionally good mastery of one’s native tongue (Meaning speaking language).
   2. I am not sure that I entirely agree with this. While I do believe that you must at least know the language you are programming in I am not sure you have to have an exceptionally good mastery of it. Now, I am not a professional programmer nor am I an amazing one so I could be wrong.
3. **Explain why you believe Dijkstra’s statement that “Projects promoting programming in ‘natural language’ are intrinsically doomed to fail,” is true or false**.
   1. I think this statement if false. I believe he is making this statement because of the steps that must be made to have the code be readable by the computer since humans and computers don’t understand in the same language and back in 1975 when this appears to have been made that process was slow. Now a days computers and programs are efficient at doing that and those languages make it much easier to learn and understand.
4. **What is the key to reliability, according to Dijkstra? Explain.**
   1. Simplicity is what Dijkstra says is the prerequisite for reliability. I believe he says this because the more complicated things get the more likely things will break and the more likely smalls things will cause much bigger things.
5. **Explain why you think his statement that “In spite of its name, software engineering requires (cruelly) hard science for its support,” is true or false.**
   1. I believe that this statement is true because without the hardware the Software would have nothing to talk to and nothing to do. There would be no Software Engineering without someone coming up with the hardware for it to live in. If there was no Logic Board in my Computer, then there would be no Operating System.
6. **Given Knuth’s statements, what do you see as the difference between a computer programmer and a computer scientist?**
   1. I think the difference is in what they do. A computer programmer uses the languages they have available to them and creates while a computer scientist is looking deeper than the surface of the languages that are used and looking at how they are being used and how they might be used in the future.
7. **Do you think Knuth’s efforts to write in a jargon-free manner is a good thing?**
   1. Yes, I do. Jargon is often used within disciplines and between collegues, but when writing instructional material, it is always good practice to leave that jargon out because you never know who might be reading it or who might need to read it and writing it jargon free gives them the best chance to understand.
8. **What do you think is the basis for Knuth’s statement about Dijkstra programming in C++?**
   1. C++ is considered a programming language in a “Nature Language” which Dijkstra said these languages are doomed to fail. So, he would not be using a language like C++.
9. **Do you see any correlation between the comment about what makes a mathematician’s life worthwhile and a computer scientist’s life worthwhile?**
   1. I honestly am not sure. I have sat with this for a while, and I cannot make a connection between this quote and the quote in the interview. Is it that they want their work to be admired?
10. **How do you see the tools you use to program, given this quote?**
    1. I believe that to be successful at programming you must enjoy the tools that you are using. Meaning you must enjoy programming in the language that you have chosen to do to the project in as well as you must enjoy the interface you are using like the keyboard and the operating system. You also must like the program that you are programming in. If you don’t like these things, then it will be hard to be good at programming.