

PART 1

1. In “Can Programming Be Liberated from the Von Neumann Style?”, Backus discussed the von Neumann bottleneck. Explain what he means by this.
 - John Backus describes the **von Neumann bottleneck** as the inefficiency caused by constant data movement between the CPU and memory. This architecture forces programs to focus on low-level data manipulation, rather than abstract problem-solving, limiting performance and flexibility. Backus argues that this model slows computation and restricts the development of more expressive programming systems.
2. Backus claims that the assignment operator “splits programming into two worlds.” Describe these “two worlds” and the advantages/disadvantages he sees in each.
 - Backus explains that the **assignment operator** divides programming into two worlds: **expressions** and **statements**. Expressions, which perform computations, have clear algebraic properties, making them easier to reason about. Statements, on the other hand, control the program’s flow and modify its state, introducing complexity and side effects. While expressions are predictable and structured, statements offer power but create disorder, making programs harder to manage.
3. From the Thomas reading, summarize what he means by transforming data and describe the main benefit Thomas sees in this approach.
 - In the Thomas reading, **transforming data** involves converting input into useful output by focusing on function-based transformations rather than managing object states. This approach, used in functional programming like Elixir, simplifies programming by reducing complexity and enabling flexibility. It supports efficient, parallel processing by allowing functions to be combined in pipelines, making code more adaptable and scalable.