CSCI330 54|Homework2

Zoljargal Enkhbayar

Q2.

2. The two common data structures that were included in the Plankalkül were arrays and records.

5. The slowness of interpretation of programs acceptable in the early 1950s due to the lack of floating-point hardware in the available computers.

6. The IBM 704's hardware support for floating-point operations made interpretive systems less viable, leading to the rise of compiled languages like Fortran, which could take advantage of the faster hardware.

7. The Fortran design project begun in November 1954.

8. The primary application area of computers at the time Fortran was designed was scientific computations.

9. The source of all the control flow statements of Fortran I was the IBM 704 instructions. These control flow statements were directly based on the machine instructions of the IBM 704 computer.

10. The most significant feature added to Fortran I to get Fortran II was the independent compilation of subroutines. This allowed subroutines to be compiled separately, so changes to a program are no longer required recompiling the entire program, making it possible to work with much larger programs.

11. Fortran 77 added logical loop control statements and an If statement with an optional Else clause, improving branching and loop flexibility over Fortran IV.

14. Linguists were interested in artificial intelligence in the late 1950s because they were focused on natural language processing, aiming to develop methods for computers to understand and process human languages.

15. Lisp was developed at MIT by John McCarthy and Marvin Minsky as part of the MIT AI Project.

20. The missing language element of ALGOL 60 that damaged its chances for widespread use was lack of input/output (I/O) facilities. This made it less practical for general programming tasks, especially in areas where interacting with the user or external systems was crucial.

21. The language designed to describe the syntax of ALGOL 60 was Backus-Naur Form (BNF). It was created by John Backus and his team to formally define the grammar of ALGOL 60.

22. COBOL was based on FLOW-MATIC, a language developed by UNIVAC for business applications. In what year did the COBOL design process begin? The COBOL design process began in 1959.

23. The COBOL design process began in 1959.

24. The record data structure in COBOL originated with Plankalkül.

25. The U.S. Department of Defense was most responsible for the early success of COBOL, particularly through its sponsorship of the COBOL design process and its widespread adoption in government applications.

36. A non-procedural language is one where you describe what you want to achieve, not how to do it. The system figures out the execution details. Examples include SQL, Prolog, and HTML.

37. In Prolog, facts are simple truths like "John is a man," and rules define how new facts can be inferred, like "If X is wise, X is a philosopher." These allow Prolog to answer queries by reasoning through them.

46. Ada and COBOL both prioritize strong typing, readability, and structured programming. Ada is used for real-time and embedded systems, while COBOL is designed for business applications. Both languages are standardized and emphasize reliability in critical environments.

51. The original version of Perl was meant to replace sh (a shell scripting language) and awk (a text processing language).

52. JavaScript is most widely used for web programming, particularly for tasks like validating form input data and dynamically creating and modifying HTML documents. It is primarily embedded in web browsers to enable interactive and dynamic functionality on websites.

57. Ruby's arithmetic operators are unique because they are syntactic mechanisms that specify method calls. In Ruby, operators like +, -, *, etc., are methods, and because they are methods, they can be redefined. This allows for more flexibility and customization compared to traditional languages where operators are usually built-in and cannot be altered.

59. Lua is impurely interpreted. It is translated to an intermediate code before being interpreted, like early Java implementations. This makes it flexible and portable.

60. The C# switch statement addresses a significant deficiency in C's switch statement by allowing the use of strings as cases, whereas C's switch can only use integers or characters. This change provides more flexibility and usability in programming.