Name:_____Group6_____ CSI 311
Discussion 2

From Wikipedia:

Plankalkül (German pronunciation: [ˈplaːnkalkyːl], "Plan Calculus") is a computer language designed for engineering purposes by Konrad Zuse between 1943 and 1945. It was the first high-level non-von Neumann programming language to be designed for a computer. Also, notes survive with scribblings about such a plan calculation dating back to 1941. Plankalkül was not published at that time owing to a combination of factors such as conditions in wartime and postwar Germany and his efforts to commercialize the Z3 computer and its successors. In 1944 Zuse met with the German logician and philosopher Heinrich Scholz and they discussed Zuse's Plankalkül. In March 1945 Scholz expressed his deep appreciation to Zuse for his utilization of the logical calculus.

By 1946, Zuse had written a book on the subject but this remained unpublished. In 1948 Zuse published a paper about the Plankalkül in the "Archiv der Mathematik" but still did not attract much feedback - for a long time to come programming a computer would only be thought of as programming with machine code. The Plankalkül was eventually more comprehensively published in 1972 and the first compiler for it was implemented in 1998. Another independent implementation followed in the year 2000 by the Free University of Berlin.[i]

Remember his syntax:

- An assignment statement to assign the expression A[4] + 1 to A[5]

  | A + 1 => A

V | 4 5 (subscripts)

S | 1.n 1.n (data types)

1. What feature of PlankalKül do you think would have had the greatest influence on Fortran if the Fortran designers had been familiar with PlankalKül? (20 points)

   If the designer had been familiar with PlankalKül, Fortran would use Boolean or single bit as primitive data type, have included arrays and records as its data structure and perform syntax an alysis on logic formulas that had parentheses and precedence.

2. Draw PlankalKül interpretation of the following statement: (20 points)

   A[2] + B[3] => A[1]

Name:_____Group6_____ CSI 311

Discussion 2

| A + B => A

V | 2 3 1 (subscripts)

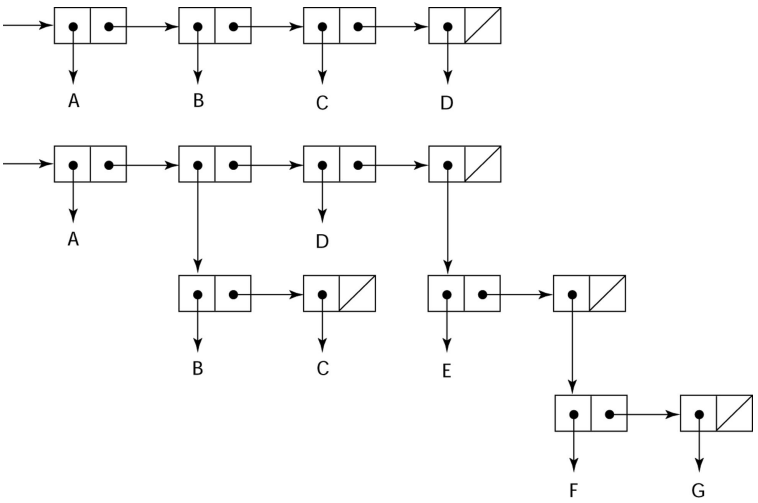
S | 1.n 1.n 1.n (data types)

LISP programming theories:

LISP:

- LISt Processing language
  - Designed at MIT by McCarthy
- AI research needed a language to
  - Process data in lists (rather than arrays)
  - Symbolic computation (rather than numeric)
- Only two data types: atoms and lists
- Syntax is based on *lambda calculus*

The following diagram is for the list: (A (B C) D (E (FG)))

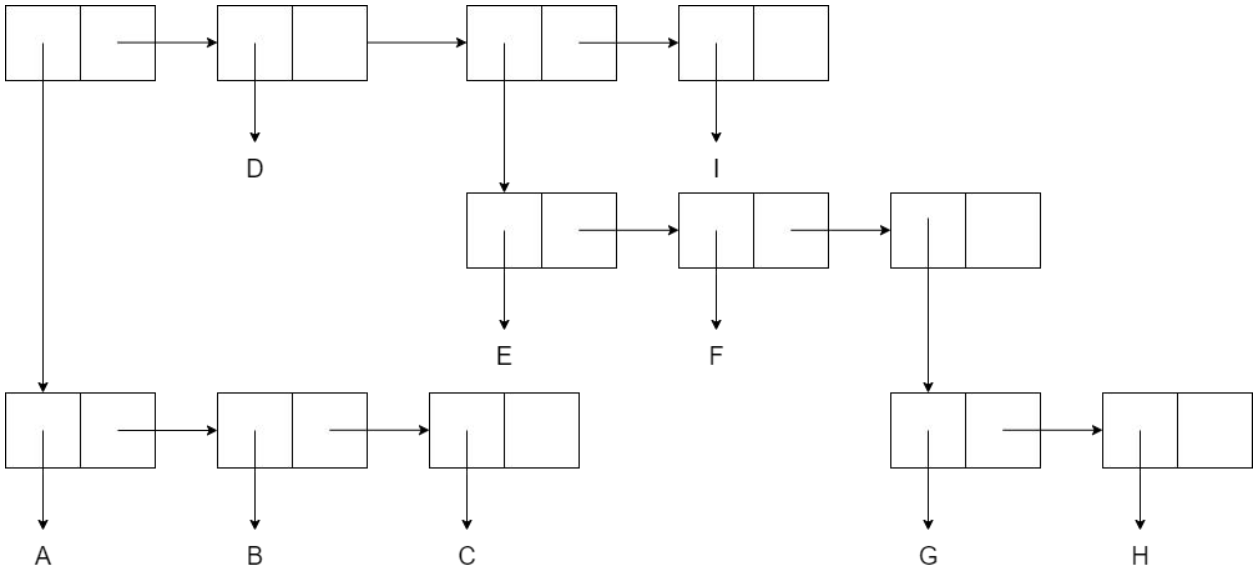


3. Make and education guess as to what might be the most common syntax error in LISP programs? (10 points)

   The most common syntax error in LISP is the Incorrect placement or number of parentheses because in LISP, program code and data have exactly the same form which is parenthesized lists.

Name:_____Group6_____ CSI 311
Discussion 2

4. Draw the diagram for the following list: (30 points)
((A B C) D (E F (G H)) I)



5. Why in your opinion, do new scripting languages appear more frequently than new compiled languages? (justify your answer) (20 points)

Scripts enable power and flexibility because they are highly extensible. Users can develop highly complex solutions for complex test cases with the power of . NET scripting. With scripts, users can leverage the functionality provided by external assemblies to enhance the functionality available within scripts.

These scripts are written to automate a task like a call to the server etc. within the major program. The scripts written are easy to learn and use.

Reference：

1.https://redirect.cs.umbc.edu/courses/331/fall04/notes/pdf/02history.pdf
2.https://history-computer.com/plankalkul-guide/
3. https://history-computer.com/plankalkul-guide/
4. https://slideplayer.com/slide/4235885/