INFORMATION PROCESSING LANGUAGE

programming language created at RAND corporation and Carniege institute of technology at about 1956 by:
allen newell : language specifier, application programmer
cliff shaw: system programmer
hebert a simon: application programmer, user

the language includes features intended to help with programs that perform simple problem solving actions such as lists, dynamic memory allocation, data types, recursion, functions as arguments, generators, and cooperative multitasking. IPL invented the concept of list-processing, albeit in an assembly language style.

paradigm: assembly stable release: IPL-V1
OS: cross-platform: JOHNNIAC, IBM 650, IBM 704, IBM 7090
influenced: lisp

IPL arguably introduced several programming language features:

- list manipulation: but only lists of atoms, not general lists
- property lists: but only when attached to other lists.
- higher-order functions: except that assembly language programming has always been able to compute with addresses of functions to call; IPL was an early attempt to generalize this property of assembly language and in a principled way.
- computation with symbols: except that the symbols are letter + number, not full words.
- virtual machine:

IPL was first utilized to demonstrate that the theorems in Principia Mathematica which were laboriously proven by hand, by Bertrand Russell and Alfred North Whitehead, could in fact be proven by computation.

IPL was used to implement several early artificial intelligence programs, also by the same authors: the Logic Theorist (1956), the General Problem Solver (1957), and their computer chess program NSS (1958).

the language was soon displaced by Lisp, which had far more powerful features, a simpler syntax, and the benefit of automatic garbage collection.