Language Review (versus C)

Data I/O

In FORTRAN, data I/O is very similar to C in a sense that the console acts like a file and it assigns an integer file-descriptor to each opened file. You can read / write to files using a predefined format string like libc's printf / scanf.
In COBOL, the console is separate from files and the files that the program uses must be defined in a special header along with the contained data format.

Arithmetic Operations

In FORTRAN arithmetic operations is similar to C, ASCII maths expressions can be evaluated in-line and data types are also similar, some operators are replaced by words due to FORTRAN's small character set but they almost behave the same.
In COBOL, all data types seem to be strings with 3 sets: [0-9], [a-zA-Z], and *. Arithmetic operations resembles that of the English language although ASCII maths expressions can also be evaluated via the COMPUTE statement.

Subroutine

Both FORTRAN and COBOL offers code organization features with subroutines. FORTRAN has C-like functions where arguments are passed by reference while all variables(fields) in COBOL are global.

DDA Implementation

The hardest part about this task is not in fact the use of ancient programming languages because they are very much full-featured and productions-ready, however the limitation of using only IF and GO TO requires some tricks to emulate higher-level constructs such as for-loops.

FORTRAN

Loops and branches can be easily implemented using labels and conditional jumps, techniques are very similar to those of assembly programming.

COBOL

Implementation in COBOL was not trivial because the language did not have labels. Instead, subroutine names are used with the GO TO statement, this will disturb the subroutine call trace and thus the executions flow as well if not used properly. (i.e. A subroutine will not return if GO TO is used to direct control elsewhere). The workaround is to have a subroutine conditionally GO TO itself to implement a loop, since the GO TO statement will not add to the call trace, this sill simply result in a flat loop.