# SPICE Circuit Solver

## Overview

This Python program is designed to read and solve electrical circuits described using the SPICE netlist format. It performs Modified Nodal Analysis to calculate node voltages and current values through voltage sources in the circuit.

## Global Variables

The program uses Global Variables to store circuit information such as the components, nodes, etc. Global variables have been used for ease of information sharing between functions, simplicity and to organize the code by separating the concerns of data storage and manipulation. The global variables related to circuit components help centralize information about the circuit. I also acknowledge that it is not the ideal way to solve this problem and that excessive use of globals can lead to code that is hard to maintain and debug, especially in larger programs.

## Algorithm

The algorithm used to performs circuit analysis using Modified Nodal Analysis for SPICE circuits is

1. **Initialization**: Initialize variables, reset the circuit state, and read the SPICE circuit file.
2. **Component Extraction**: Parse the file to extract component details, categorize and store them lists and dictionaries, and identify unique nodes. Raise an error if necessary
3. **Matrix Preparation**: Build modified admittance and constants matrices based on component values and connections.
4. **Solve Circuit**: Solve the circuit equations using linear algebra. Raise an error if the matrix is singular (no solution).
5. **Output Generation**: Create dictionaries to store node voltages and current values.
6. **Return Results**: Provide the calculated node voltages and current values as output.

## Usage

1. Import the `evalSpice` module in your Python script.
2. Use the `evalSpice(filename)` function to analyze a SPICE circuit file with `.ckt` as its extension and obtain the results in the form of a `Tuple` containing the Node Voltages and Currents through Voltage Sources in two separate dictionaries.

## Error Handling

The program raises specific exceptions with descriptive error messages when encountering file-related issues or circuit analysis problems.

- `FileNotFoundError`: Raised when the specified circuit file is not found or has an incorrect extension.
- `ValueError`: Raised when the circuit file is malformed, extra components are included in the definition, or there is no solution to the circuit.

## Test Cases

The given test cases were handled along with a few additional test cases

- Circuits that do not contain a 'GND' node
- Circuits with Current Sources in Series
- Various forms of Malformed Circuit files
- Negative Resistances
- Files with the wrong extension
- Zero Resistances, i.e., Short Circuits without Combining the Nodes
    - This test case is included in the "extra" folder along with the expected output

## Modules and References

The NumPy and OS modules were used in this program. The implementation of Modified Nodal Analysis was done by reading through and using the essence of the following MNA 1, MNA 2 and implementing it for only resistive circuits as specified in the problem statement.