Extra Credit Project: CMOS Logic Gate SPICE Generator

Write C++ code to generate the CMOS logic gate SPICE description

Describe the MOS transistor as follows:

```
Mname drain gate source body type [W=x  L=y]
```

Mname is a name you pic, however, it must start by an upper case M. type is either NMOS or PMOS. For the purpose of this project always connect the body to the source. You can also optionally provide information about W and L (the dimensions of each transistors where W/L is the aspect ratio).

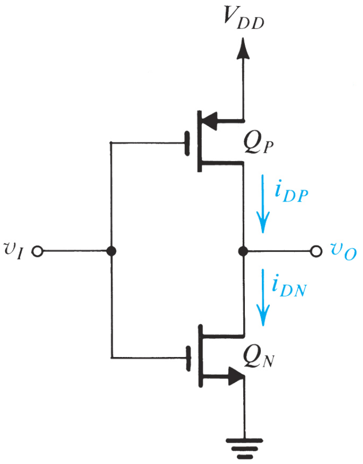
For user input use &, |, and ' , for and, or, and not. The user can input any Boolean function using any symbols. However, the output symbol cannot be used as an input symbol. Do not allow spaces in the middle of the equation.

For example: y=a&b|c'   is a valid input
                        y=y|a   is not valid

The program can run in one of two modes: (1) The simple mode, (2) The transistor-scaling mode. You do not have to implement both modes.

(1) The simple mode (2% Extra credit) => Just generate the CMOS circuit without sizing information.

If the user inputs
vo=vi'



The output should be

```
M1  vo vi vdd vdd PMOS
M2  vo  vi gnd gnd NMOS
```

(2) The transistor-scaling mode (2% extra credit) => Adding sizing information to make the delay of the gate close to that of a CMOS inverter with sizes p and n (as done in class).

The user inputs
vo=vi’ L=2 n=2 p=4

The output should be

```
M1  vo vi vdd vdd PMOS W=8 L=2
M2  vo  vi gnd gnd NMOS W=4 L=2
```

Instructions:

1) You can work in groups of 2 to 3 students.
2) When you run the program, the program should wait for the user to input the Boolean function (and optionally the sizing information in the order shown in the second example). Your program should automatically detect the mode of operation.
3) The output should be displayed on the screen and saved in a text file. You should print the PUN first followed by the PDN. Always assume that we have a voltage source called vdd and a ground called gnd.
4) Write a report explaining all your implementation details including the data structures and all algorithms used. Moreover, clearly state any assumptions.
5) Write a detailed testing plan that includes pairs of Boolean functions and the expected output.
6) Do not use any libraries related to windows or mac. Ensure your code will run on any platform.
7) Submit both the report and your C++ code.
8) The grading rubric:
    a. 20% on the clarity of the report
    b. 80% on the correct functionality (we will run random test cases and check if your code produced the correct output).