ASSIGNMENT 2
Program Synthesis using Symbolic Execution

ANSHUL PUROHIT (231110007)

IMPLEMENTATION:

- Initialize a Z3 Solver:
  - I've created an instance of the Z3 solver by calling z3.z3Solver(). This solver will be used to check for symbolic equivalence between two programs.
- Load Test Data:
  - It opens and reads two JSON files containing test data for two programs. These JSON files represent sets of test cases.
- Comparison of Test Data:
  - I've iterated through the keys in both JSON files. For each pair of test cases with the same parameter values (as determined by comparing the 'params' field).
  - I've retrieved the 'params', and 'symEnc' fields from both test data sets.
  - I've replaced single quotes (') with double quotes (") in the 'params' field using .replace("'", "\""), which is used to ensure valid JSON formatting.
  - I've evaluated the symbolic encodings ('symEnc') from both test data sets as Python expressions using eval.
  - I've iterated through the keys in parameter_1, which represent parameters in the test data, and added each parameter as a symbolic variable to the Z3 solver using solver.addSymbVar().
  - After adding all the constraints, I used the Z3 solver to check whether the constraints were satisfiable by calling solver.s.check(). The result (answer) will be one of "sat" (satisfiable) or "unsat" (unsatisfiable).
  - Depending on the result (answer), I've printed a message to the console. If the constraints are satisfiable (i.e., "sat"), it indicates that the programs are symbolically equivalent and also prints the model that satisfies the constraints using solver.s.model(). If the constraints are unsatisfiable (i.e., "unsat"), it indicates that the programs are not symbolically equivalent.

ASSUMPTIONS:

I've assumed that the structure of the program should be somewhat the same.

LIMITATIONS:

The limitation of this approach is that it'll work for the cases where we will give the same input for both programs.

Another limitation is that it'll work for the cases where constant parameters aren't present in the conditional statements.