Assignment #2: Program Synthesis using Symbolic Execution

Name: Yashvir Singh Nathawat

Roll No: 231110059

IMPLEMENTATION:

I have implemented this complex task using following basic steps:

1) **Run Symbolic Execution**
   a) Run Symbolic Execution on both files one with hole and other without hole.
   b) It generates two .json files namely testData.json and testData_2.json
2) **Extract Json Files**
   a) Extract two json files in testData and testData_2.
   b) Extracts params and symbEnc from file P2(without hole) and constraints and symbEnc from file P1(with hole)
3) **Match Constraints.**
   a) Use Z3 solver(s1) to inputs from P2 to file in Constraints from P1
   b) If any constraint satisfies, then equate symbEnc of both file P1 and P2 and pass this constraint to Z3 Solver (s)
4) **Z3 Solvers:**
   a) **Solver s1** to match inputs and constraints and equate symbEnc of both file to **Solver s**.
   b) **Solver s** solves the final constraint.
5) **Getting the result.**

a) If the condition are satisfiable using solver.model() function we are getting the required constraints.

Assumptions:

The provided code has several assumptions:

1. No constant parameters should be given while running P2.
2. The value of the output variables is used to determine whether two programs are equal.
3. It is assumed that the JSON data in **testData.json** and **testData_2.json** is well-formed and valid. This includes assuming that all required fields exist and have the expected data types.

Limitations:

The provided code has several limitations:

1. When there is a constant in the condition that impacts the control flow of the response, this program will not function.
2. The code does not include comprehensive error handling. For example, it does not account for cases where files cannot be opened, JSON parsing fails, or other potential exceptions.