Plan for training

1. Create classes such as Adder, Subtractor, Multiplier, Divider, etc. to implement basic math functions and write BDD test cases for them.
   1. Addition (+)
   2. Subtraction (-)
   3. Multiplication (\*)
   4. Division (/)
   5. Modulo (%)
   6. Or (||)
   7. And (&&)
   8. Not (!)
   9. Comparison operators (==, !=, >, <, >=, <=)
   10. Bitwise operators (~, &, |, ^)
   11. Increment and decrement operators
   12. +=, -=, \*=, /= and %=
   13. Math library functions such as exp, log, sin, cos, etc. (Over 50 functions)
   14. Basic data structures (Arrays, Lists, Deques, Queues, etc.)
2. Have to train both generative and discriminative networks so we have to write both correct test cases and incorrect test cases to train these networks.
3. The discriminative network needs to learn the target language’s syntax in order to validate the test cases. We need correctly and incorrectly created test code for this end. Since there is no requirement for the syntax to be learnt via test cases strictly, therefore, we can use all sorts of valid and invalid code for training the discriminative network.