


## Input:

```
CodeOptimization.py
1  # Importing Modules
2  import re
3
4  class CodeOptimizer:
5      def Intermediate_Code(self, grammar):
6          intermediate_code = []
7          for rule in grammar:
8              lhs, rhs = rule.split('=')
9              variables = re.findall(r'[A-Z]', rhs)
10             operators = re.findall(r'[+~*/]', rhs)
11             converted_rhs = tuple(operators + variables + [lhs])
12             intermediate_code.append(converted_rhs)
13         return intermediate_code
14
15     def optimize_code(code):
16         var_dict = {}
17         new_code = []
18         for op, var1, var2, var3 in code:
19             key = f"{var1}{op}{var2}"
20             if key in var_dict:
21                 new_tuple = ('0', var_dict[key], '0', var3)
22                 new_code.append(new_tuple)
23             else:
24                 var_dict[key] = var3
25                 new_code.append((op, var1, var2, var3))
26         return new_code
27
28 # Example input grammar
29 grammar = [
30     "A=B+C",
31     "B=A-D",
32     "C=D+E",
33     "D=B+C",
34     "E=A-D",
35     "F=D+E"
36 ]
37
38 # The Grammar Before Optimization
39 print("\n The entered Grammar is:")
40 for rule in grammar:
41     print(" ", rule, end="\n")
42 print()
43
44 # Initializing a code_optimizer object
45 code_optimizer = CodeOptimizer()
46
47 # Intermediate Code Generation
48 intermediate_code = code_optimizer.Intermediate_Code(grammar)
49
50 # Optimized Intermediate Code Generation
51 optimized_code = CodeOptimizer.optimize_code(intermediate_code)
52
53 # The Grammar After Optimization
54 print(" The grammar after optimization is: ")
55
56 # Loop through each tuple in the code and print it in the desired format
57 for op, var1, var2, var3 in optimized_code:
58     if op == '+':
59         print(" ", f"{var3} = {var1} + {var2}")
60     elif op == '-':
61         print(" ", f"{var3} = {var1} - {var2}")
62     elif op == '*':
63         print(" ", f"{var3} = {var1} * {var2}")
64     elif op == '/':
65         print(" ", f"{var3} = {var1} / {var2}")
66     else:
67         print(" ", f"{var3} = {var1}")
```



```
"C:\\Users\\Karthik Shetty\\Desktop\\C\\Python\\venv\\Scripts\\python.exe"

The entered Grammar is:
A=B+C
B=A-D
C=B+C
D=A-D

The grammar after optimization is:
A = B + C
B = A - D
C = A
D = B
```