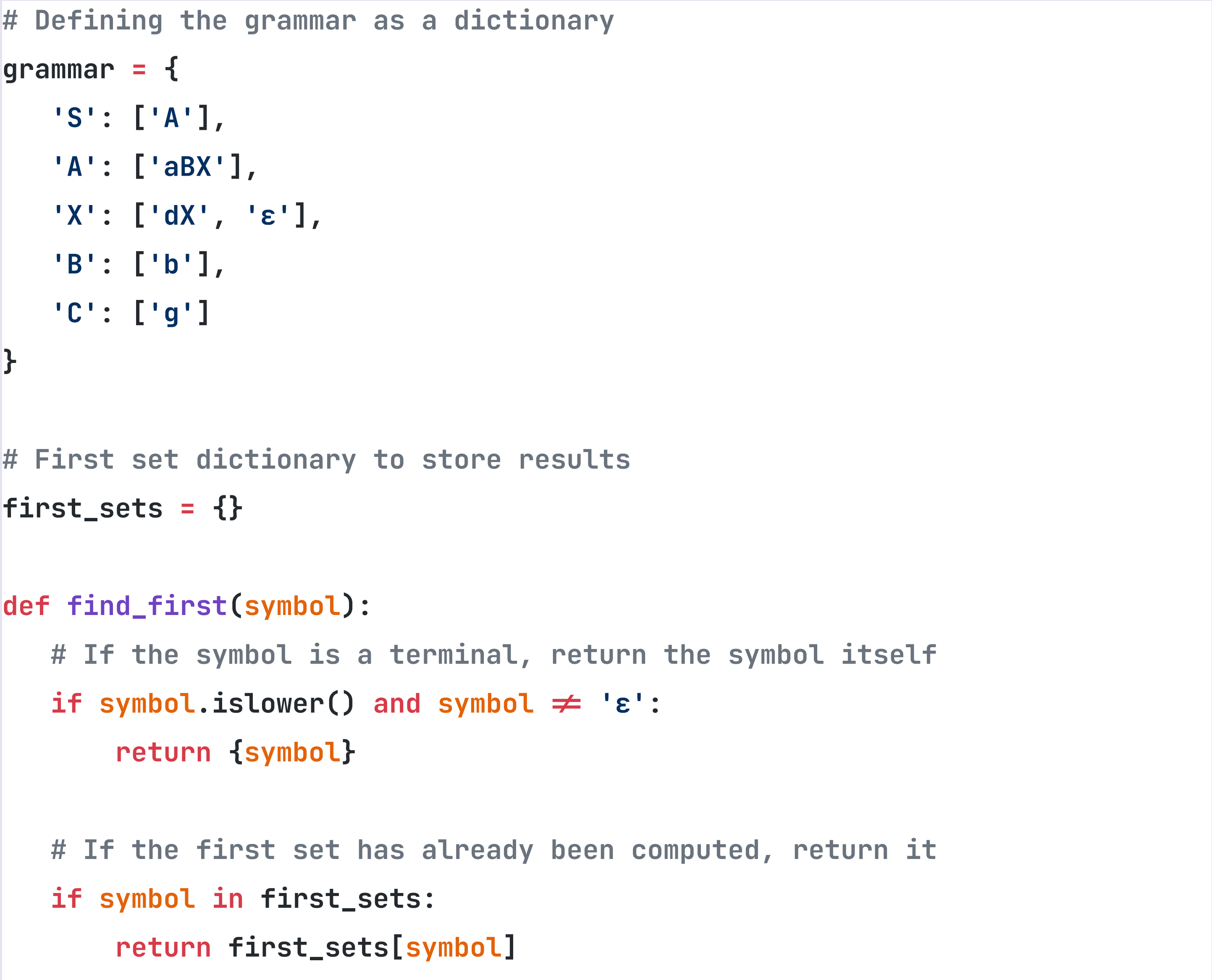
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 71
CD Practical 6

Aim : Write a Program to find First of Given Grammar.

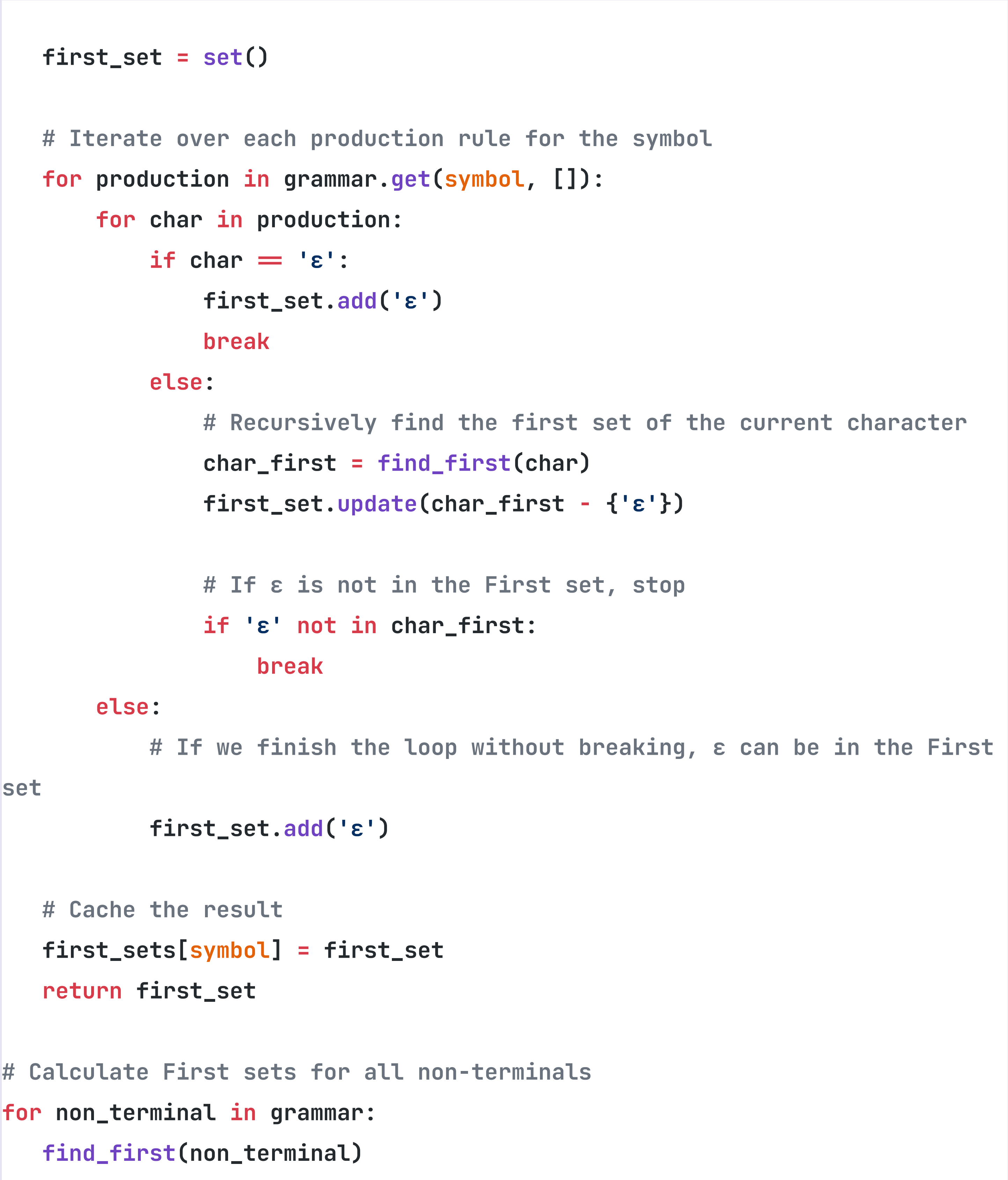
S -> A
A -> aBX
X -> dX | ε
B -> b
c -> g

Code :



```python
# Defining the grammar as a dictionary
grammar = {
    'S': ['A'],
    'A': ['aBX'],
    'X': ['dX', 'ε'],
    'B': ['b'],
    'C': ['g']
}

# First set dictionary to store results
first_sets = {}

def find_first(symbol):
    # If the symbol is a terminal, return the symbol itself
    if symbol.islower() and symbol ≠ 'ε':
        return {symbol}

    # If the first set has already been computed, return it
    if symbol in first_sets:
        return first_sets[symbol]
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 71
CD Practical 6

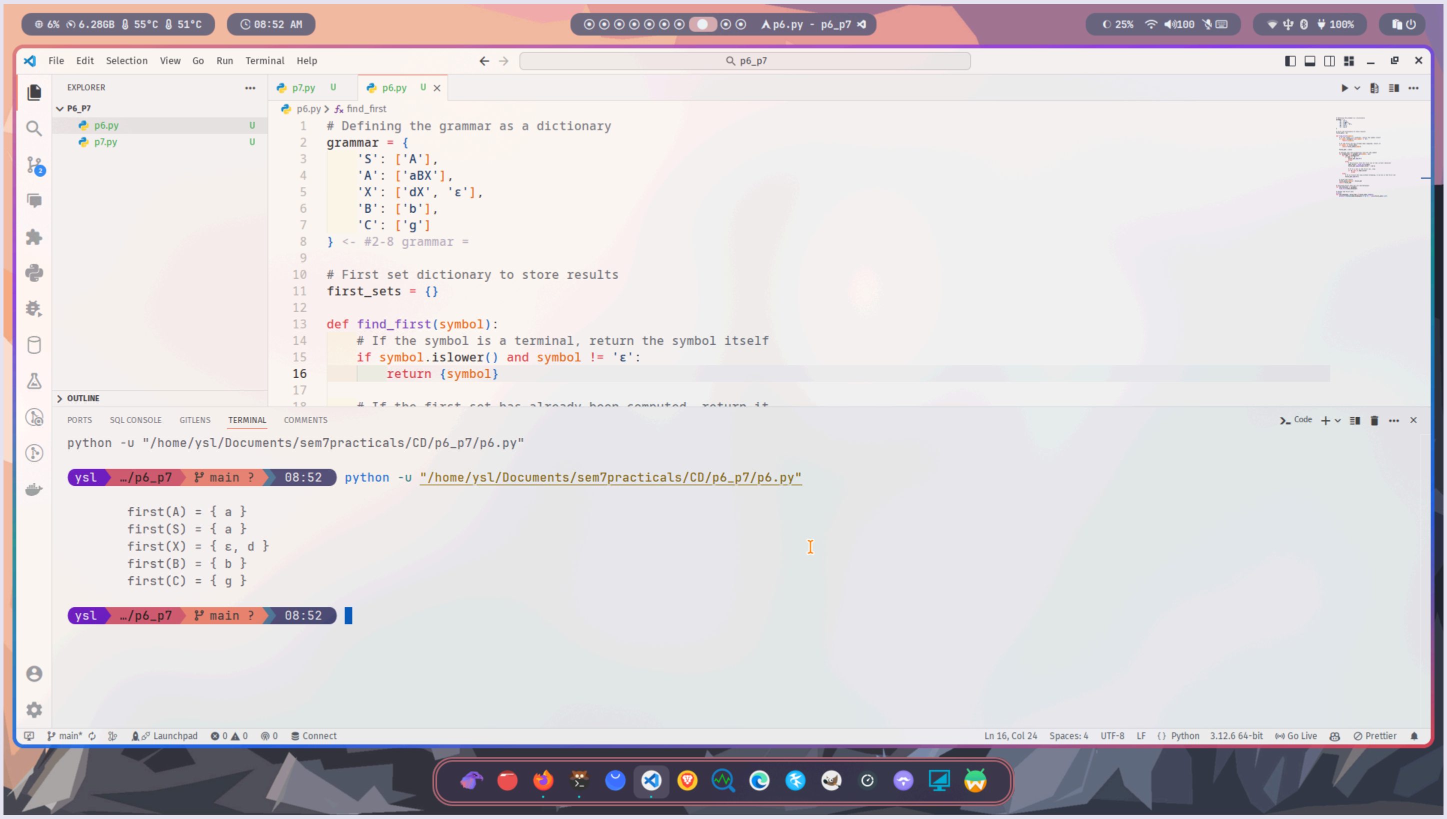


```
   first_set = set()

   # Iterate over each production rule for the symbol
   for production in grammar.get(symbol, []):
       for char in production:
           if char == 'ε':
               first_set.add('ε')
               break
           else:
               # Recursively find the first set of the current character
               char_first = find_first(char)
               first_set.update(char_first - {'ε'})

               # If ε is not in the First set, stop
               if 'ε' not in char_first:
                   break
       else:
           # If we finish the loop without breaking, ε can be in the First
set
           first_set.add('ε')

   # Cache the result
   first_sets[symbol] = first_set
   return first_set

# Calculate First sets for all non-terminals
for non_terminal in grammar:
   find_first(non_terminal)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 71
CD Practical 6

```
# Output the First sets
print()
for non_terminal, first_set in first_sets.items():
    print(f"\tfirst({non_terminal}) = {{ {', '.join(first_set)} }}")
```

Output :



```
# Defining the grammar as a dictionary
grammar = {
    'S': ['A'],
    'A': ['aBX'],
    'X': ['dX', 'ε'],
    'B': ['b'],
    'C': ['g']
} <- #2-8 grammar =

# First set dictionary to store results
first_sets = {}

def find_first(symbol):
    # If the symbol is a terminal, return the symbol itself
    if symbol.islower() and symbol != 'ε':
        return {symbol}

    # If the first set has already been computed, return it
```

```
python -u "/home/ysl/Documents/sem7practicals/CD/p6_p7/p6.py"

ysl  …/p6_p7  main ?  08:52  python -u "/home/ysl/Documents/sem7practicals/CD/p6_p7/p6.py"

        first(A) = { a }
        first(S) = { a }
        first(X) = { ε, d }
        first(B) = { b }
        first(C) = { g }

ysl  …/p6_p7  main ?  08:52
```