NAME: KSHITIJ GUPTA
Enrolment Number: 21162101007
Sub: CD

Practical – 6[Batch-71]

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]
```

```python
    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)
```
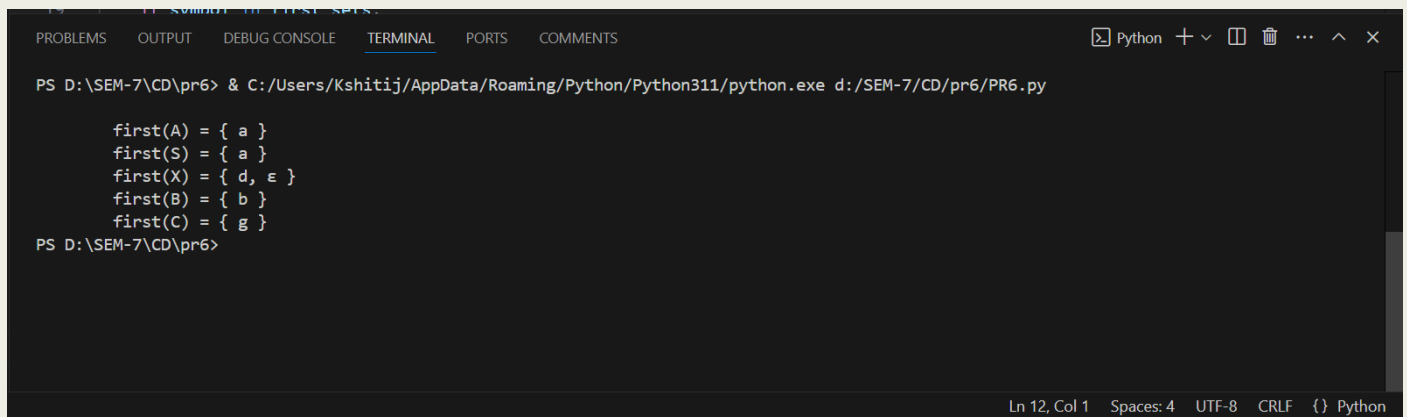
 # Output the First sets

print()

for non_terminal, first_set in first_sets.items():

   print(f"\tfirst({non_terminal}) = {{ {', '.join(first_set)} }}")

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    COMMENTS                                    >_ Python  + ∨  ⊡  🗑  ···  ∧  ✕

PS D:\SEM-7\CD\pr6> & C:/Users/Kshitij/AppData/Roaming/Python/Python311/python.exe d:/SEM-7/CD/pr6/PR6.py

        first(A) = { a }
        first(S) = { a }
        first(X) = { d, ε }
        first(B) = { b }
        first(C) = { g }
PS D:\SEM-7\CD\pr6>
```

Ln 12, Col 1   Spaces: 4   UTF-8   CRLF   {} Python