Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: CD(Compiler Design)

Branch: CBA

Batch:71

----------PRACTICAL 06----------

Write a Program to find First of Given Grammar.
S -> A
A -> aBX
X -> dX | ε
B -> b
c -> g

✓ 6.py :

```
# Grammar definition
productions = {
    'S': ['A'],
    'A': ['aBX'],
    'X': ['dX', 'ε'],  # 'ε' represents the empty string
    'B': ['b'],
    'c': ['g']
}

# Dictionary to store first sets
first_sets = {}

# Function to calculate the First of a non-terminal
def find_first(non_terminal):
    # If First set is already computed, return it
    if non_terminal in first_sets:
        return first_sets[non_terminal]

    first_set = set()  # To store the First of the non-terminal
    productions_for_non_terminal = productions.get(non_terminal, [])
```

```
    for production in productions_for_non_terminal:
        for symbol in production:
            if symbol.islower():  # If it's a terminal, add it to the
First set
                first_set.add(symbol)
                break
            elif symbol == 'ε':  # If epsilon, add it to the First set
                first_set.add('ε')
                break
            else:  # If it's a non-terminal, recursively calculate its
First
                first_of_symbol = find_first(symbol)
                first_set.update(first_of_symbol - {'ε'})

                # If First of the symbol contains ε, check the next
symbol
                if 'ε' not in first_of_symbol:
                    break
        else:
            # If we reach here, all symbols can derive ε, so add ε to
the First set
            first_set.add('ε')

    # Store and return the computed First set
    first_sets[non_terminal] = first_set
    return first_set

# Calculate First sets for all non-terminals
def compute_first_sets():
    for non_terminal in productions:
        find_first(non_terminal)

# Main function to run the program
if __name__ == '__main__':
    compute_first_sets()
    # Output the First sets
    for non_terminal, first in first_sets.items():
        print(f"First({non_terminal}) = {{ {', '.join(first)} }}")
```

✓ **Output :**