FF:

from collections import defaultdict

def compute\_first():

while True:

changed = False

for nt in rules:

for prod in rules[nt]:

tokens = prod.split()

temp = set()

for tok in tokens:

temp.update(first[tok])

if "ε" not in first[tok]: # Stop if no epsilon

break

else:

temp.add("ε")

if temp - first[nt]: # If new elements found

first[nt].update(temp)

changed = True

if not changed:

break

def compute\_follow():

follow[start\_symbol].add("$")

while True:

changed = False

for nt in rules:

for prod in rules[nt]:

tokens = prod.split()

for i, tok in enumerate(tokens):

if tok in rules:

temp = set()

for next\_tok in tokens[i+1:]:

temp.update(first[next\_tok])

if "ε" not in first[next\_tok]:

break

else:

temp.update(follow[nt])

temp.discard("ε")

if temp - follow[tok]: # If new elements found

follow[tok].update(temp)

changed = True

if not changed:

break

def display\_results():

print("\nFIRST Sets:")

for nt in rules:

print(f"{nt}: {sorted(first[nt])}")

print("\nFOLLOW Sets:")

for nt in rules:

print(f"{nt}: {sorted(follow[nt])}")

# ---------- Main Program ---------- #

rules = defaultdict(list)

first = defaultdict(set)

follow = defaultdict(set)

# User Inputs

terminals = input("Enter terminals (space-separated): ").split()

non\_terminals = input("Enter non-terminals (space-separated): ").split()

start\_symbol = non\_terminals[0]

print("Enter productions (format: A -> alpha | beta), type 'done' to stop:")

while True:

inp = input().strip()

if inp.lower() == "done":

break

head, body = inp.split("->")

rules[head.strip()] = [p.strip() for p in body.split("|")]

# Initialize FIRST sets

for t in terminals:

first[t] = {t}

for nt in non\_terminals:

first[nt] = set()

compute\_first()

compute\_follow()

display\_results()