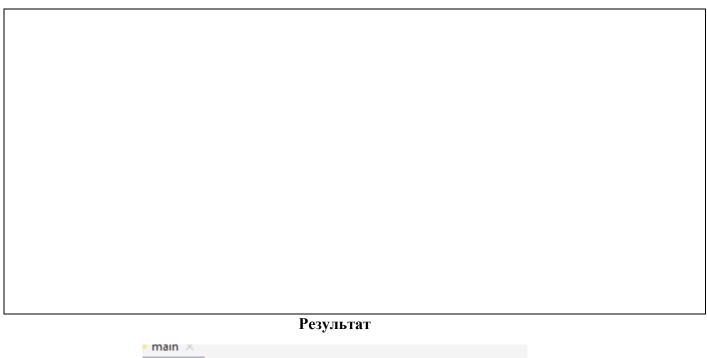
Лабораторна робота №6.3

Задано формулу з використанням логічних операцій (див. 6.1). Перетворіть формулу на еквівалентну, в якій використовується тільки Штрих Шеффера (або Стрілка Пірса).

```
Кол
def evaluate_formula(formula, variable_values):
   Обчислює значення логічної формули з використанням заданих значень змінних.
   Аргументи:
    - formula (str): Рядок, що представляє логічну формулу з логічними операціями (*
-AND, +-OR, --NAND).
   - variable values (dict): Словник, що містить значення змінних у форматі змінна:
значення.
   Повертає:
    - int: Результат обчислення формули (0 або 1).
   output queue = []
   operator stack = []
   precedence = {'*': 2, '+': 1, '-': 1}
   for char in formula:
       if char.isalpha():
           output queue.append(variable values[char])
       elif char == '0':
           output queue.append(0)
       elif char == '1':
            output queue.append(1)
        elif char in ['*', '+', '-']:
            while operator stack and operator stack[-1] != '(' and precedence[char]
<= precedence[operator stack[-1]]:</pre>
                output queue.append(operator stack.pop())
            operator stack.append(char)
        elif char == '(':
            operator stack.append(char)
        elif char == ')':
            while operator stack and operator stack[-1] != '(':
                output queue.append(operator stack.pop())
            operator stack.pop()
   while operator stack:
        output queue.append(operator stack.pop())
   stack = []
    for token in output queue:
        if token == '-':
            operand = stack.pop()
            stack.append(int(not operand))
        elif token == '*':
            operand2 = stack.pop()
           operand1 = stack.pop()
           stack.append(int(operand1 and operand2))
        elif token == '+':
            operand2 = stack.pop()
```

```
operand1 = stack.pop()
            stack.append(int(operand1 or operand2))
        else:
            stack.append(token)
    return stack.pop()
def replace operators (formula):
    Замінює символи логічних операцій в формулі на слова.
    Аргументи:
    - formula (str): Рядок, що містить логічну формулу з символами операцій (*, +, -
) .
    Повертає:
    - str: Рядок заміненої формули з операціями у словесному форматі (AND, OR, NAND).
    replacements = {'*': ' AND ', '+': ' OR ', '-': ' NAND '}
    for old, new in replacements.items():
        formula = formula.replace(old, new)
    return formula
def check equivalence(formula1, formula2, variable values):
    Перевіряє еквівалентність двох логічних формул з використанням заданих значень
змінних.
    Аргументи:
    - formulal (str): Рядок, що представляє першу логічну формулу.
    - formula2 (str): Рядок, що представляє другу логічну формулу.
    - variable values (dict): Словник, що містить значення змінних у форматі змінна:
значення.
    Повертає:
    - bool: True, якщо формули еквівалентні, False - якщо ні.
    result1 = evaluate formula(formula1, variable values)
    result2 = evaluate formula(formula2, variable values)
    return result1 == result2
formula1 = "A - C"
formula2 = "B"
variable values = {'A': 1, 'B': 0, 'C': 1}
converted formula1 = replace operators(formula1)
converted formula2 = replace operators(formula2)
equivalent = check equivalence(formula1, formula2, variable values)
print(f"Значення змінних: {variable values}")
print(f"Формула 1: {converted formula1}")
print(f"Формула 2: {converted formula2}")
if equivalent:
   print("Формули еквівалентні")
else:
    print("Формули не еквівалентні")
```



```
"H:\University\2 курс\2 семестр\Дискретні Струк
Значення змінних: {'A': 1, 'B': 0, 'C': 1}
Формула 1: А NAND C
Формула 2: В
Формули еквівалентні

Process finished with exit code 0

"H:\University\2 курс\2 семестр\Дискретні Струк
Значення змінних: {'A': 1, 'B': 0, 'C': 1}
Формула 1: А NAND В
Формула 2: В
Формула 2: В
Формули не еквівалентні

Process finished with exit code 0
```