МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное автономное образовательное учреждение высшего образования «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБО6РОСТРОЕНИЯ»

	I	КАФЕДРА № 43	
ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКО	٦й		
	JY1		
ПРЕПОДАВАТЕЛЬ			
Старший преподав			С.А. Рогачев
должность, уч. степень,	звание	подпись, дата	инициалы, фамилия
ОТЧЕТ О ЛАБОРАТО	ЭРНОЙ РАБОТЕ	5 No3	
01 121 0 11 12 011 11	711101111111111111111111111111111111111	11.20	
		ФС Поста	
	_	_	
	по курсу: Теор	оия Вычислительных Про	цессов
РАБОТУ ВЫПОЛНИЛ			
СТУДЕНТ ГР. №	4134к		Н.А. Костяков
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Построить формальную систему Поста FSp, реализующую вычисление заданной арифметической функции. Написать программу на языке высокого уровня имитирующую (эмулирующую) вычисления на основе выводимости в формальной системе Поста. [⊕] Программа должна работать на любых входных данных из заданного множества. [⊕] Программа должна удовлетворять предъявляемым требованиям.

Постановка задачи;

Реализовать функцию деления на 10 (x/10).

Листинг программы на языке высокого уровня с комментариями;

```
class PostSystem:
        init__(self, initial_string, productions, alphabet):
    self.string = initial string # Начальная строка
    self.productions = productions # Множество продукций (правил)
   self.alphabet = alphabet # Алфавит символов
 def apply production(self, production):
    # Проверяем, можно ли применить правило
    lhs, rhs = production
    if lhs in self.string:
      # Применяем правило и возвращаем результат
      self.string = self.string.replace(lhs, rhs, 1)
      return True
   return False
  def run(self, output_file):
    with open(output_file, 'w', encoding='utf-8') as f:
      while True:
        f.write(f"Текущая строка: {self.string}√n")
        applied = False
        for production in self.productions:
          # Пробуем применить правило
          if self.apply production(production):
            f.write(f"Применено правило: {production[0]} -> {production[1]}\n")
            applied = True
            break
        # Если ни одно правило не было применено, заканчиваем
        if not applied:
          f.write("Достигнут конец вывода\n")
          break
def read_file(filename):
 with open(filename, 'r') as f:
   return f.read().strip()
```

```
def read_productions(filename):
 productions = []
 with open(filename, 'r') as f:
    for line in f:
      lhs, rhs = line.strip().split("->")
      productions.append((lhs.strip(), rhs.strip()))
 return productions
def main():
  # Читаем начальную строку, алфавит и правила
  initial_string = read_file('input_string.txt')
  alphabet = read_file('alphabet.txt').split()
  productions = read productions('productions.txt')
  # Проверка символов начальной строки
 for symbol in initial_string:
    if symbol not in alphabet:
      raise Exception(f"Символ '{symbol}' не в алфавите")
 # Запуск системы Поста
  post_system = PostSystem(initial_string, productions, alphabet)
 post_system.run('output.txt')
if__name__ == "__main__":
 main()
```

Содержимое входного файла согласно заданию;

Примеры результатов выполнения:

Применено правило: 0000000000 -> 1

Применено правило: 0000000000 -> 1 Текущая строка: 110000000000000000 Применено правило: 0000000000 -> 1

Текущая строка: 1110000000 Применено правило: 0000000 ->

Текущая строка: 111 Достигнут конец вывода

Символ '3' не в алфавите

Вывод

Была построена программа по формальной системе Поста