Лабораторная работа 8

Попов Дмитрий Павлович, НФИмд-01-23

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра математического моделирования и искусственного интеллекта

ПРЕЗЕНТАЦИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Математические основы защиты информации и информационной безопасности

Преподователь: Кулябов Дмитрий Сергеевич
Студент: Попов Дмитрий Павлович
Группа: НФИмд-01-23

МОСКВА
2023 г.

Прагматика выполнения лабораторной работы

Прагматика выполнения лабораторной работы

Требуется реализовать:

1. Алгоритм сложения неотрицательных целых чисел
2. Алгоритм вычитания неотрицательных целых чисел
3. Алгоритм умножения неотрицательных целых чисел столбиком
4. Алгоритм деления многоразрядных целых чисел

Цель работы

Цель работы

Освоить на практике целочисленную арифметику многократной точности.

Выполнение лабораторной работы

1. Алгоритм сложения неотрицательных целых чисел основан на стандартном методе сложения в столбик.

```python
def add_non_negative_numbers(u, v, base):
    n = max(len(u), len(v))
    u = [int(digit) for digit in u.zfill(n)][::-1]
    v = [int(digit) for digit in v.zfill(n)][::-1]

    result = [0] * n
    carry = 0

    for j in range(n):
        Wj = u[j] + v[j] + carry
        result[j] = Wj % base
        carry = Wj // base

    return ''.join(map(str, result[::-1]))
```

Figure 1: add

2. Алгоритм вычитания неотрицательных целых чисел основан на стандартном методе вычитания в столбик.

```
def subtract_non_negative_numbers(u, v, base):
    n = max(len(u), len(v))
    u = [int(digit) for digit in u.zfill(n)][::-1]
    v = [int(digit) for digit in v.zfill(n)][::-1]

    result = [0] * n
    borrow = 0

    for j in range(n):
        Wj = u[j] - v[j] - borrow
        if Wj < 0:
            Wj += base
            borrow = 1
        else:
            borrow = 0
        result[j] = Wj
```

3. Алгоритм умножения неотрицательных чисел столбиком базируется на стандартном методе умножения в столбик.

```python
def multiply_non_negative_numbers(u, v):
    len_u, len_v = len(u), len(v)
    result = [0] * (len_u + len_v)

    for i in range(len_u - 1, -1, -1):
        carry = 0
        for j in range(len_v - 1, -1, -1):
            temp = int(u[i]) * int(v[j]) + carry + result[i + j + 1]
            result[i + j + 1] = temp % 10
            carry = temp // 10
        result[i] += carry

    result_str = ''.join(map(str, result))
    return result_str.lstrip('0') or '0'
```

Figure 3: mult

4. Алгоритм деления многоразрядных целых чисел основан на делении в столбик.(1/2)

```
def divide_large_numbers(dividend, divisor):
    # Преобразование строк в списки цифр
    dividend = [int(digit) for digit in str(dividend)]
    divisor = [int(digit) for digit in str(divisor)]

    quotient = []  # Частное
    remainder = 0  # Остаток

    for digit in dividend:
        current_dividend = remainder * 10 + digit
        current_quotient = current_dividend // divisor[0]
        remainder = current_dividend % divisor[0]
        quotient.append(current_quotient)
```

Figure 4: div1

4. Алгоритм деления многоразрядных целых чисел основан на делении в столбик.(2/2)

```
    # Проход по остальным разрядам
    for i in range(len(dividend) - len(divisor) + 1):
        current_quotient = quotient[i]
        current_remainder = remainder
        j = 1

        while j < len(divisor) and j + i < len(dividend):
            current_dividend = current_remainder * 10 + dividend[i + j]
            current_quotient = current_dividend // divisor[j]
            current_remainder = current_dividend % divisor[j]
            j += 1

        quotient[i] = current_quotient
        remainder = current_remainder

    # Приведение к строке
    quotient_str = ''.join(map(str, quotient)).lstrip('0') or '0'
    remainder_str = str(remainder)

    return quotient_str, remainder_str
```

Figure 5: div2

5. Вывод программы:

```
C:\Users\79119\AppData\Local\Programs\Python\Python310\python.exe C:/Users/79119/PycharmProjects/Inform_security_labs/src/lab_8.py
Введите первое неотрицательное число: 1421351
Введите второе неотрицательное число: 31
Введите основание системы счисления: 10
Сумма чисел 1421351 и 31 в системе счисления с основанием 10 равна 1421382
Разность чисел 1421351 и 31 в системе счисления с основанием 10 равна 1421320
Произведение чисел 1421351 и 31 в системе счисления с основанием 10 равно 44061881
Частное и остаток от деления чисел 1421351 и 31 в системе счисления с основанием 10 равны ('24213513', '0')

Process finished with exit code 0
```

Figure 6: output

Выводы

Выводы

В результате выполнения работы я освоил на практике дискретное логарифмирование в конечном поле.