

# **Математические основы защиты информации и информационной безопасности. Лабораторная работа №8**

**Целочисленная арифметика многократной точности**

Студент: Лесков Данила Валерьевич НФИмд-02-21

Преподаватель: Кулябов Дмитрий Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задачи</b>	<b>6</b>
<b>3</b>	<b>Теоретические сведения</b>	<b>7</b>
3.1	Арифметические операции . . . . .	7
<b>4</b>	<b>Выполнение работы</b>	<b>8</b>
4.1	Реализация алгоритмов . . . . .	8
4.2	Пример работы алгоритмов . . . . .	13
<b>5</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

# List of Figures

4.1	Пример работы . . . . .	13
-----	-------------------------	----

## List of Tables

# 1 Цель работы

Изучение целочисленной арифметики для больших чисел.

## 2 Задачи

Реализовать программно алгоритмы сложения неотрицательных целых чисел, вычитания неотрицательных целых чисел, умножения неотрицательных целых чисел столбиком, быстрого столбика и деления многоразрядных целых чисел.

## 3 Теоретические сведения

Для арифметических операций над большими целыми числами в системе счисления  $b$ , где  $b$  - натуральное число,  $b \geq 2$ , применяется запись этого числа посимвольно. Натуральное  $b$ -разрядное число будем записывать в виде

$$u = u_1 u_2 \dots u_n$$

### 3.1 Арифметические операции

В данной работе операции по работе с большими целыми числами производится в некоторой системе счисления  $b$ . Предусматривается реализация алгоритмов сложения, вычитания, умножения и деления. На вход каждой из функций подается два числа и система счисления, на выходе возвращается результат арифметической операции.

Подробнее об арифметических алгоритмах: [2].

## 4 Выполнение работы

### 4.1 Реализация алгоритмов

```
def algorithm1(u, v, n, b):  
    j = n  
    k = 0  
    w = [None for i in range(j + 1)]  
    while True:  
        w[j] = (u[j - 1] + v[j - 1] + k) % b  
        j = j - 1  
        if j > 0:  
            continue  
        elif j == 0:  
            w[0] = k  
            break  
    return w
```

```
def algorithm2(u, v, n, b):  
    j = n  
    k = 0  
    w = [None for i in range(j)]  
    while True:
```



```

    w[j - 1] = (u[j - 1] - v[j - 1] + k) % b
    j = j - 1
    if j > 0:
        continue
    elif j == 0:
        break
return w

```

```

def algorithm3(u, v, b):
    m = len(v) - 1
    n = len(u) - 1
    w = [0 for i in range(n + m + 1)]
    j = m
    while True:
        if (v[j] == 0):
            w[j] = 0
        else:
            i = n
            k = 0
            while True:
                t = u[i] * v[j] + w[i + j] + k
                w[i + j] = t % b
                k = t / b
                i = i - 1
                if i > 0:
                    continue
            else:
                w[j] = k

```

```

        break
    j = j - 1
    if j > 0:
        continue
    elif j == 0:
        break
return w

```

```

def algorithm4(u, v, b):
    t = 0
    m = len(v)
    n = len(u)
    w = [0 for i in range(n + m + 1)]
    for s in range(m + n):
        for i in range(s):
            t = t + (u[n - i - 1] * v[m - s + i - 1])
        w[m + n - s] = int(t % b)
        t = t / b
    return w

```

```

def algorithm5(u, v, b):
    n = len(u)
    t = len(v)
    q = [0 for j in range(n - t + 1)]
    while u >= v * (b ** (n - t)):
        q[n - t] += 1
        u = u - v * (b ** (n - t))

```

```

    for i in range(n, t + 1):
        if (u[i] >= v[t]):
            q[i - t - 1] = b - 1
        else:
            q[i - t - 1] = ((u[i] * b) + u[i - 1]) / v[t]
            while q[i - t - 1] * ((v[t] * b) + v[t -
1]) > u[i] * (b ** 2) + u[i - 1] * b + u[i - 2]:
                q[i - t - 1] -= 1
            u = u - q[i - t - 1] * (b ** (i - t - 1)) * v
            if u < 0:
                u = u + v * (b ** (i - t - 1))
                q[i - t - 1] -= 1

    r = u
    return q, r

```

```

if __name__ == '__main__':
    while True:
        try:
            result_code = int(input(
                """

```

Выберите алгоритм:

- 1 - Сложение неотрицательных целых чисел;
- 2 - Вычитание неотрицательных целых чисел;
- 3 - Умножение неотрицательных целых чисел столбиком;
- 4 - Быстрый столбик;
- 5 - Деление многоразрядных целых чисел

-----

0 - Выход из программы

Введите номер операции: ""

```
    ))
    if result_code > 5:
        print("Ошибка ввода!")
        continue
    if result_code == 0:
        break
except:
    print("Ошибка ввода!")
    continue
arr1 = input("Введите первое целое число: ")
arr1 = list(arr1)
arr2 = input("Введите второе целое число: ")
arr2 = list(arr2)
system = int(input("Введите систему счисления 2..16: "))
arr1 = list(map(lambda x: int(x), arr1))
arr2 = list(map(lambda x: int(x), arr2))

if result_code == 1:
    print(algorithm1(arr1, arr2, len(arr1), system))

if result_code == 2:
    print(algorithm2(arr1, arr2, len(arr1), system))

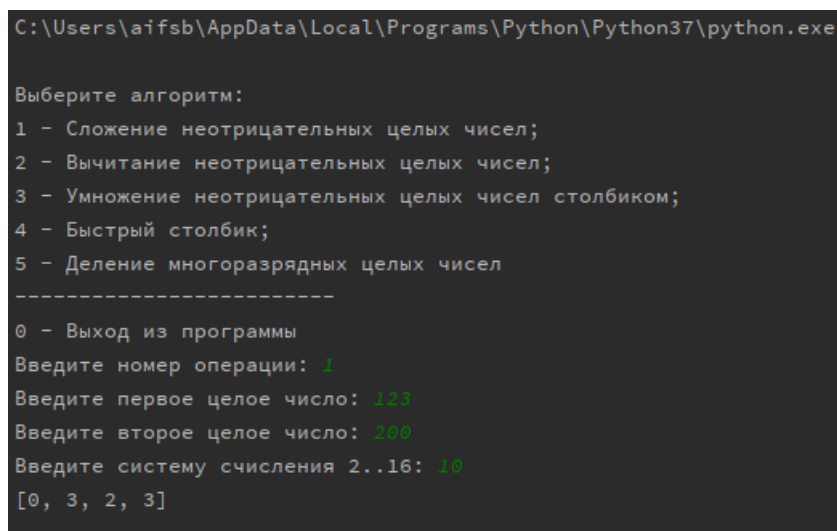
if result_code == 3:
    print(algorithm3(arr1, arr2, system))

if result_code == 4:
    print(algorithm4(arr1, arr2, system))
```

```
if result_code == 5:  
    print(algorithm5(arr1, arr2, system))
```

## 4.2 Пример работы алгоритмов

На рис. 4.1 представлен пример взаимодействия пользователя с алгоритмами через консольное меню:



```
C:\Users\ai\fsb\AppData\Local\Programs\Python\Python37\python.exe  
  
Выберите алгоритм:  
1 - Сложение неотрицательных целых чисел;  
2 - Вычитание неотрицательных целых чисел;  
3 - Умножение неотрицательных целых чисел столбиком;  
4 - Быстрый столбик;  
5 - Деление многоразрядных целых чисел  
-----  
0 - Выход из программы  
Введите номер операции: 1  
Введите первое целое число: 123  
Введите второе целое число: 200  
Введите систему счисления 2..16: 10  
[0, 3, 2, 3]
```

Figure 4.1: Пример работы

## 5 Выводы

В ходе выполнения работы была успешно изучена целочисленная арифметика для больших чисел. Были программно реализованы алгоритмы сложения неотрицательных целых чисел, вычитания неотрицательных целых чисел, умножения неотрицательных целых чисел столбиком, быстрого столбика и деления многоразрядных целых чисел.

## Список литературы

1. Большие числа [Электронный ресурс]. Википедия, 2021. URL: [https://ru.wikipedia.org/wiki/Большие\\_числа](https://ru.wikipedia.org/wiki/Большие_числа).
2. Длинная арифметика [Электронный ресурс]. Википедия, 2021. URL: [https://ru.wikipedia.org/wiki/Длинная\\_арифметика](https://ru.wikipedia.org/wiki/Длинная_арифметика).