

# **Отчет по лабораторной работе №8**

**Дисциплина: Математические основы защиты информации и  
информационной безопасности**

**Выполнила Дяченко Злата Константиновна, НПМмд-02-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
4.1	Шаг 1 . . . . .	9
4.2	Шаг 2 . . . . .	9
4.3	Шаг 3 . . . . .	10
4.4	Шаг 4 . . . . .	10
4.5	Шаг 5 . . . . .	11
<b>5</b>	<b>Выводы</b>	<b>12</b>

## Список иллюстраций

4.1	Реализация алгоритма сложения неотрицательных чисел . . . . .	9
4.2	Реализация алгоритма вычитания неотрицательных чисел . . . .	10
4.3	Реализация алгоритма умножения неотрицательных целых чисел столбиком . . . . .	10
4.4	Реализация алгоритма быстрого столбика . . . . .	11
4.5	Реализация алгоритма деления многоразрядных целых чисел . .	11

# 1 Цель работы

Ознакомится и реализовать алгоритмы целочисленной арифметики многократной точности.

## 2 Задание

Реализовать программно пять алгоритмов: алгоритм сложения неотрицательных чисел, алгоритм вычитания неотрицательных чисел, алгоритм умножения неотрицательных чисел, алгоритм быстрого столбика, алгоритм деления много-разрядных целых чисел.

### 3 Теоретическое введение

Алгоритм сложения неотрицательных чисел.

*Вход.* Два неотрицательных числа  $u = u_1 u_2 \dots u_n, v = v_1 v_2 \dots v_n$ ; разрядность чисел  $n$ ; основание системы счисления  $b$ .

*Выход.* Сумма  $w = w_0 w_1 \dots w_n$ , где  $w_0$  – цифра переноса – всегда равная 0 либо 1.

1. Присвоить  $j := n, k := 0$  ( $j$  идет по разрядам,  $k$  следит за переносом).
2. Присвоить  $w_j = (u_j + v_j + k)(mod b)$ , где  $w_j$  – наименьший неотрицательный вычет в данном классе вычетов;  $k = \lfloor \frac{u_j + v_j + k}{b} \rfloor$ .
3. Присвоить  $j := j - 1$ . Если  $j > 0$ , то возвращаемся на шаг 2; если  $j = 0$ , то присвоить  $w_0 := k$  и результат:  $w$ .

Алгоритм вычитания неотрицательных чисел.

*Вход.* Два неотрицательных числа  $u = u_1 u_2 \dots u_n, v = v_1 v_2 \dots v_n, u > v$ ; разрядность чисел  $n$ ; основание системы счисления  $b$ .

*Выход.* Разность  $w = w_1 w_2 \dots w_n = u - v$ .

1. Присвоить  $j := n, k := 0$  ( $k$  заем из старшего разряда).
2. Присвоить  $w_j = (u_j - v_j + k)(mod b)$ , где  $w_j$  – наименьший неотрицательный вычет в данном классе вычетов;  $k = \lfloor \frac{u_j - v_j + k}{b} \rfloor$ .
3. Присвоить  $j := j - 1$ . Если  $j > 0$ , то возвращаемся на шаг 2; если  $j = 0$ , то присвоить  $w_0 := k$  и результат:  $w$ .

Алгоритм умножения неотрицательных целых чисел столбиком.

*Вход.* Числа  $u = u_1 u_2 \dots u_n, v = v_1 v_2 \dots v_m$ ; основание системы счисления  $b$ .

*Выход.* Произведение  $w = uv = w_1 w_2 \dots w_{m+n}$  1. Выполнить присвоения:  $wt + 1 := 0, wt + 2 := 0, \dots, wt + n := 0, j \leftarrow m$  ( $j$  перемещается по номерам

разрядов числа  $v$  от младших к старшим).

2. Если  $v_j = 0$ , то присвоить  $w_j := 0$  и перейти на шаг 6.

3. Присвоить  $i \leftarrow n, k \leftarrow 0$  (Значение  $i$  идет по номерам разрядов числа  $u$ ,  $k$  отвечает за перенос).

4. Присвоить  $t := u_i * v_j + w_{i+j} + k, w_{i+j} := t \bmod b, k := \frac{t}{b}$ , где  $w_{i+j}$  – наименьший неотрицательный вычет в данном классе вычетов.

5. Присвоить  $i \leftarrow i - 1$ . Если  $i > 0$ , то возвращаемся на шаг 4, иначе присвоить  $w_j := k$ .

6. Присвоить  $j \leftarrow j - 1$ . Если  $j > 0$ , то вернуться на шаг 2. Если  $j = 0$ , то результат  $w$ .

Алгоритм быстрого столбика.

Вход. Числа  $u = u_1 u_2 \dots u_n, v = v_1 v_2 \dots v_m$ ; основание системы счисления  $b$ .

Выход. Произведение  $w = uv = w_1 w_2 \dots w_{m+n}$

1. Присвоить  $t := 0$ .

2. Для  $s$  от 0 до  $m+n-1$  с шагом 1 выполнить шаги 3 и 4.

3. Для  $i$  от 0 до  $s$  с шагом 1 выполнить присвоение  $t := t + u_{[n-i]} * v_{[m-s+i]}$ . 4. Присвоить  $w_{m+n-s} := t \bmod b, t := \frac{t}{b}$ , где  $w_{m+n-s}$  – наименьший неотрицательный вычет по модулю  $b$ . Результат:  $w$ .

Алгоритм деления многоразрядных целых чисел.

Вход. Числа  $u = u_n \dots u_1 u_0, v = v_t \dots v_1 v_0, n \geq t \geq 1, v_t \neq 0$ ; разрядность чисел соответственно  $n$  и  $t$ .

Выход. Частное  $q = q_{n-t} \dots q_0$ , остаток  $r = r_t \dots r_0$ .

1. Для  $j$  от 0 до  $n-t$  присвоить  $q_j := 0$ .

2. Пока  $u \geq v b^{n-t}$ , выполнять  $q_{n-t} := q_{n-t} + 1, u := u - v b^{n-t}$ .

3. Для  $i = n, n-1, \dots, t+1$  выполнить пункты 3.1-3.4.

3.1 если  $u_i \geq v_t$ , то присвоить  $q_{i-t-1} := b - 1$ , иначе присвоить  $q_{i-t-1} := \frac{u_i b + u_{i-1}}{v_t}$

3.2 пока  $q_{i-t-1}(v_t b + v_{t-1}) > u_i b^2 + u_{i-1} b + u_{i-2}$  выполнять  $q_{i-t-1} := q_{i-t-1} - 1$ .

3.3 присвоить  $u := u - q_{i-t-1} b^{i-t-1} v$

3.4 если  $u < 0$ , то присвоить  $u := u + v b^{i-t-1}, q_{i-t-1} := q_{i-t-1} - 1$ .

4.  $r:=u$ . Результат:  $q$  и  $r$ .



## 4 Выполнение лабораторной работы

### 4.1 Шаг 1

Ознакомилась с предоставленными теоретическими данными. Для выполнения задания решила использовать язык Python. Написала функцию, реализующую алгоритм сложения неотрицательных чисел. Код функции и результат ее использования представлен на Рисунке 1 (рис. - fig. 4.1). Функция принимает на вход числа  $u$ ,  $v$ ,  $b$ . Пример работы алгоритма для числа из представленных для лабораторной работы материалов также представлен на рисунке.

```
In [54]: def alg_1(u, v, b):  
         u=[int(a) for a in str(u)]  
         v=[int(a) for a in str(v)]  
         n=len(u)  
         k=0  
         w=[0]*(n+1)  
         for j in range (n-1, -1, -1):  
             w[j+1]=(u[j]+v[j]+k)%b  
             k=math.floor((u[j]+v[j]+k)/b)  
         w[0]=math.floor(k)  
         print(w)  
  
In [55]: alg_1(125,125,10)  
[0, 2, 5, 0]  
  
In [56]: alg_1(125000,125000,10)  
[0, 2, 5, 0, 0, 0, 0]
```

Рис. 4.1: Реализация алгоритма сложения неотрицательных чисел

### 4.2 Шаг 2

Реализовала алгоритм вычитания неотрицательных чисел, написав функцию. Код функции и результат ее использования представлен на Рисунке 2 (рис. - fig. 4.2).

```

In [57]: def alg_2(u,v,b):
        if (u>v):
            u=[int(a) for a in str(u)]
            v=[int(a) for a in str(v)]
            n=len(u)
            k=0
            w=[0]*n
            for j in range (n-1, -1, -1):
                w[j]=(u[j]-v[j]+k)%b
                k=math.floor((u[j]-v[j]+k)/b)
            print(w)
        else:
            print("Первое число должно быть больше второго")

In [61]: alg_2(125,100,10)
        [0, 2, 5]

In [63]: alg_2(125000,100000,10)
        [0, 2, 5, 0, 0, 0]

```

Рис. 4.2: Реализация алгоритма вычитания неотрицательных чисел

### 4.3 Шаг 3

Реализовала алгоритм умножения неотрицательных целых чисел столбиком, написав функцию. Код функции и результат ее использования представлен на Рисунке 3 (рис. - fig. 4.3).

```

In [83]: def alg_3(u,v,b):
        u=[int(a) for a in str(u)]
        v=[int(a) for a in str(v)]
        n=len(u)
        m=len(v)
        w=[0]*(n+m)
        for j in range(m-1, -1, -1):
            if (v[j]==0):
                w[j]=0
                continue
            k=0
            for i in range(n-1, -1, -1):
                t=u[i]*v[j]+w[i+j+1]+k
                w[i+j+1]=t%b
                k=math.floor(t/b)
            w[j]=k
        print(w)

In [85]: alg_3(11,11,10)
        [0, 1, 2, 1]

In [86]: alg_3(11000,11000,10)
        [0, 1, 2, 1, 0, 0, 0, 0, 0, 0]

```

Рис. 4.3: Реализация алгоритма умножения неотрицательных целых чисел столбиком

### 4.4 Шаг 4

Реализовала алгоритм быстрого столбика. Код функции и результат ее применения представлен на Рисунке 4 (рис. - fig. 4.4).

```
In [96]: def alg_4(u,v, b):
        u=[int(a) for a in str(u)]
        v=[int(a) for a in str(v)]
        n=len(u)
        m=len(v)
        w=[0]*(n+m)
        t=0
        for s in range(0, m+n-1, 1):
            for i in range(0, s):
                t=t+u[n-i-1]*v[m-s+i-1]
            w[m+n-s-1]=t%b
            t=math.floor(t/b)
        print(w)

In [98]: alg_4(1000, 1500, 10)

[0, 1, 5, 0, 0, 0, 0, 0]
```

Рис. 4.4: Реализация алгоритма быстрого столбика

## 4.5 Шаг 5

Реализовала алгоритм деления многоразрядных целых чисел. Код функции и результат ее применения представлен на Рисунке 5 (рис. - fig. 4.5).

```
In [144]: def alg_5(u,v,b):
        u_p=[int(a) for a in str(u)]
        v_p=[int(a) for a in str(v)]
        n=len(u_p)-1
        t=len(v_p)-1
        if (n>t and t>1 and v_p[0]!=0):
            q=[0]*(n-t+1)
            while (u>v*math.pow(b, (n-t))):
                q[n-t]=q[n-t]+1
                uu=v*math.pow(b, (n-t))
            u=int(u)
            u_p=[int(a) for a in str(u)]
            v_p=[int(a) for a in str(v)]
            n=len(u_p)-1
            t=len(v_p)-1
            for i in range(n, t+1-1, -1):
                if (u_p[i]>v_p[t]):
                    q[i-t+1]=b-1
                else:
                    q[i-t+1]=(u_p[i]*b+u_p[i-1])/v_p[t]
                    while ((q[i-t+1]*(v_p[t]*b+v_p[t-1]))>(u_p[i]*b+u_p[i-1]*b+u_p[i-2])):
                        q[i-t+1]=q[i-t+1]-1
                    u=int(''.join(map(str, u_p)))
                    v=int(''.join(map(str, v_p)))
                    uu=q[i-t+1]*math.pow(b, (i-t+1))*v
                    if (uu>0):
                        uu=v*math.pow(b, (i-t+1))
                        q[i-t+1]=q[i-t+1]-1
            r=uu
            q.reverse()
            print(''.join(map(str, q)))
            print(q, r)
        else:
            print("Неверно введены числа")

In [145]: alg_5(100, 10, 10)

10 0
```

Рис. 4.5: Реализация алгоритма деления многоразрядных целых чисел

## 5 Выводы

Я ознакомилась с алгоритмами целочисленной арифметики многократной точности и реализовала их программно. Результаты работы находятся в репозитории на GitHub, а также есть скринкаст выполнения лабораторной работы.