

# Отчёт по лабораторной работе 8

Супонина Анастасия Павловна

## Содержание

|                            |   |
|----------------------------|---|
| Цель работы .....          | 1 |
| Задание .....              | 1 |
| Выполнение работы .....    | 1 |
| Сложение.....              | 1 |
| Вычитание .....            | 3 |
| Умножение.....             | 3 |
| упрощенное умножение ..... | 4 |
| Деление .....              | 5 |
| Выводы.....                | 6 |
| Список литературы.....     | 7 |

## Список иллюстраций

No table of figures entries found.

## Список таблиц

No table of figures entries found.

## Цель работы

Научиться писать программы для целочисленной арифметики многократной точности.

## Задание

Реализовать следующие алгоритмы для выполнения арифметических операции: - сложение - вычитание - умножение - упрощенное умножение - деление

## Выполнение работы

### Сложение

Задаю начальные значения

n = 3 u = 222 v = 600 b = 10

Создаю функцию для нахождения суммы столбиком

```
function sum_(n,u,v,b)
    j = n
    k = 0
    w = []
    # Создаю цикл в котором беру последние цифры наших чисел
    for j in n:-1:1
        u_j = u % b
        v_j = v % b
        # В переменную w, записываю значение их суммы
        push!(w, (u_j+v_j+k) % b)
        # В переменную k, записываю значение переноса на следующий разряд,
        # использую для округления в меньшую сторону функцию floor
        k = floor((u_j+v_j+k) / b)
        # убираю из чисел последние элементы над которыми уже провела
        # операцию сложения
        u = div(u, b)
        v = div(v, b)
    end
    # При сложении мы можем из двух трехзначных чисел получить четырехзначное,
    # если при сложении последние оставшиеся числа в нашей программе, а именно
    # первые числа в исходных значениях, дадут сумму больше 9, поэтому записываю в
    # значение w0, k и если оно будет равно одному то значим так и случилось и его
    # нужно добавить к списку w и только потом преобразовывать w в результат
    w0 = k
    if w0 == 1
        push!(w, w0)
        # Так как сложение в столбик идет с конца числа, то итоговый
        # результат в переменной w у нас записан в обратном направлении и при это ещё
        # является массивом, поэтому, чтобы в выводе получить число, создаю функцию for
        result = 0
        for i in n:-1:0
            result += (10 ^ i) * w[j+1]
            j -= 1
        end
        return result
    else
        result = 0
        for i in n-1:-1:0
            result += (10 ^ i) * w[j]
            j -= 1
        end
        return result
    end
end
w = sum_(n,u,v,b)
```

```
println(w)
```

## Вычитание

```
u = 555
```

```
v = 132
```

# Записываю функцию для нахождения разности в столбик, от предыдущей функции отличаются только формулы для w и k

```
function sub_(n,u,v,b)
    j = n
    k = 0
    w = []
    for j in n:-1:1
        u_j = u % b
        v_j = v % b
        push!(w, (u_j-v_j+k) % b)
        k = floor((u_j-v_j+k) / b)
        u = div(u, b)
        v = div(v, b)
    end
    result = 0
    for i in n-1:-1:0
        result += (10 ^ i) * w[j]
        j -= 1
    end
    return result
end
w = sub_(n,u,v,b)
```

```
println(w)
```

```
u = [5; 0; 0]
```

```
v = [4; 5]
```

```
n = 3
```

```
m = 2
```

```
b = 10
```

## Умножение

Создаю функцию для умножения чисел в столбик

```
function mul(u, v, n, m, b)
    w = zeros{Int64, 1, m+n}
    j = m
    while j > 0
        if v[j] == 0
```

```

        w[j] == 0
    else
        i = n
        k = 0

        while i > 0
            t = u[i] * v[j] + w[i+j] + k
            w[i+j] = t % b
            k = div(t, b)
            i = i - 1
        end
        w[j] = k
    end
    j -= 1
end
k = n + m - 1
if w[1] == 0
    w = w[2:n+m]
    k -= 1
end
result = 0
for i in 1:1:k
    result += w[i] * (10 ^ k)
    k -= 1
end
return result
end
end

```

```

w = mul(u, v, n, m, b)
println(w)

```

```

u = [5, 0, 0]
v = [4, 5]
n = 3
m = 2
b = 10

```

## упрощенное умножение

```

function fast_mul(u, v, n, m, b)

```

```

    w = zeros{Int, m + n}
    f = m + n - 1
    for s in 0:f
        t = 0
        for i in 0:s
            ui = n - i
            vi = m - s + i
            if ui >= 1 && ui <= n && vi >= 1 && vi <= m
                t += u[n - i] * v[m - s + i]
            end
        end
        w[s+1] = t % b
        t = div(t, b)
    end
    return w
end

```

```

        end
    end

    z = (m + n) - s
    t += w[z]
    w[z] = t % b
    k = div(t, b)
    if z > 1
        w[z - 1] += k
    end
end

k = n + m - 1
if w[1] == 0
    w = w[2:n+m]
    k -= 1
end
result = 0
for i in 1:1:k
    result += w[i] * (10 ^ k)
    k -= 1
end
return result
end

println(fast_mul(u, v, n, m, b))

```

## Деление

```

u = [5, 0, 0]
v = [2, 5]
n = 3
t = 2
b = 10

function del(u, v, n, t, b)
    # преобразовываю массивы в число
    u_scalar = sum(u[i] * b^(n - i) for i in 1:n)
    v_scalar = sum(v[i] * b^(t - i) for i in 1:t)

    # Задаю значения для q
    q = zeros{Int, n - t + 1}

    while u_scalar >= v_scalar * b^(n - t)
        q[n - t + 1] += 1
        u_scalar -= v_scalar * b^(n - t)
    end

    for i in n:-1:(t + 1)
        # Вычисляю коэффициент для текущей позиции
    end
end

```

```

        q_index = i - t
        if u_scalar >= v_scalar * b^(q_index - 1)
            q[q_index] = div(u_scalar, v_scalar * b^(q_index - 1))
        else
            q[q_index] = 0
        end

        u_scalar -= q[q_index] * v_scalar * b^(q_index - 1)

        # Проверяю, чтобы u_scalar был неотрицательным
        if u_scalar < 0
            u_scalar += v_scalar * b^(q_index - 1)
            q[q_index] -= 1
        end
    end

    # записываю значение остатка
    r = u_scalar

    return q, r
end

q, r = del(u, v, n, t, b)

function res(q)
    result = 0
    l = 0
    for i in q
        result += i * (10 ^ l)
        l += 1
    end
    return result
end

println("Частное: ", res(q))
println("Остаток: ", r)

```

## Выводы

В процессе выполнения работы, я реализовала алгоритмы сложения, вычитания, умножения двух типов и деления. Таким образом я научилась создавать алгоритмы для выполнения арифметических операций с большими целыми числами на языке программирования Julia.

## Список литературы

::: Пособие по лабораторной работе 8

{[https://esystem.rudn.ru/pluginfile.php/2368425/mod\\_folder/content/0/lab08.pdf?forcedownload=1](https://esystem.rudn.ru/pluginfile.php/2368425/mod_folder/content/0/lab08.pdf?forcedownload=1)}