

# Лабораторная работа №8

## Математические основы защиты информации и информационной безопасности

Данилова Анастасия Сергеевна

### Содержание

Цель работы .....	1
Задание .....	1
Теоретическое введение .....	1
Выполнение лабораторной работы .....	2
Выводы .....	6
Список литературы .....	6

### Цель работы

Изучить алгоритмы целочисленной арифметики многократной точности и реализовать их программно на языке программирования Julia.

### Задание

- Изучить теоретическую часть о предложенных алгоритмах;
- Реализовать алгоритмы программно.

### Теоретическое введение

#### Целочисленная арифметика многократной точности

Мы считаем, что числа записаны в  $b$ -ичной системе счисления, где  $b$  — фиксированное натуральное число,  $b \geq 2$ . При этом натуральное число, записываемое не более чем  $n$  цифрами в  $b$ -ичной системе счисления, мы обозначаем  $u_1 \dots u_n$  (допуская, что несколько старших разрядов  $u_1, \dots, u_k$  могут равняться нулю). Основание  $b$  не всегда равно 2; иногда оно соответствует размеру машинного слова, отведенному под запись обычных целых чисел. В этом случае мы работаем с массивом, содержащим большое целое число. При работе с большими целыми числами удобно хранить знак такого числа в отдельной ячейке или переменной. Если мы хотим, например, перемножить два числа, то знак произведения мы вычисляем отдельно.

Рассмотрим 5 алгоритмов:

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

## Выполнение лабораторной работы

```
1 function plus(a::Vector{Int}, b::Vector{Int})
2     result = Int[]
3     carry = 0
4     max_length = max(length(a), length(b))
5
6     for i in 1:max_length
7         digit_a = i <= length(a) ? a[end - i + 1] : 0
8         digit_b = i <= length(b) ? b[end - i + 1] : 0
9         sum = digit_a + digit_b + carry
10        push!(result, sum % 10)
11        carry = div(sum, 10)
12    end
13
14    if carry > 0
15        push!(result, carry)
16    end
17
18    return reverse(result)
19 end
20 a = [1, 2, 3, 5, 4, 2]
21 b = [4, 5, 6, 4, 2, 7]
22 sum_res = plus(a, b)
23 println("Сумма: ", sum_res)
```

Сложение неотрицательных целых чисел

- **Activating** new project at `C:\Users\nastd\.julia\environments\v1.11`  
Сумма: [5, 7, 9]  
\* Terminal will be reused by tasks, press any key to close it.
- **Activating** new project at `C:\Users\nastd\.julia\environments\v1.11`  
Сумма: [5, 7, 9, 9, 6, 9]  
\* Terminal will be reused by tasks, press any key to close it.

Результат

```

1  function minus(a::Vector{Int}, b::Vector{Int})
2      result = Int[]
3      borrow = 0
4
5      for i in 1:length(a)
6          digit_a = a[end - i + 1]
7          digit_b = i <= length(b) ? b[end - i + 1] : 0
8          diff = digit_a - digit_b - borrow
9
10         if diff < 0
11             diff += 10
12             borrow = 1
13         else
14             borrow = 0
15         end
16
17         push!(result, diff)
18     end
19
20     while length(result) > 1 && result[end] == 0
21         pop!(result)
22     end
23
24     return reverse(result)
25 end
26 b = [1, 2, 3, 5, 4, 2]
27 a = [4, 5, 6, 6, 5, 7]
28 min_res = minus(a, b)
29 println("Разность: ", min_res)

```

Вычитание неотрицательных целых чисел

```

• Activating new project at `C:\Users\nastd\.julia\environments\v1.11`
Разность: [3, 3, 3, 1, 1, 5]
* Terminal will be reused by tasks, press any key to close it.

```

Результат

```

1 function multiply(a::Vector{Int}, b::Vector{Int})
2     result = zeros{Int, length(a) + length(b)}
3
4     for i in 1:length(b)
5         carry = 0
6         for j in 1:length(a)
7             product = b[end - i + 1] * a[end - j + 1] + result[end - (i + j - 2)] + carry
8             result[end - (i + j - 2)] = product % 10
9             carry = div(product, 10)
10        end
11        result[end - (i + length(a) - 1)] += carry
12    end
13
14    while length(result) > 1 && result[1] == 0
15        result = result[2:end]
16    end
17
18    return result
19 end
20 b = [1, 2, 3]
21 a = [4, 5, 6]
22 mul_res = multiply(a, b)
23 println("Произведение: ", mul_res)

```

Умножение неотрицательных целых чисел столбиком

```

● Activating new project at `C:\Users\nastd\.julia\environments\v1.11`
Произведение: [5, 6, 0, 8, 8]
✱ Terminal will be reused by tasks, press any key to close it.

```

Результат

```

1  function multiply(a::String, b::String)::String
2      digit_a = reverse(parse.(Int, collect(a)))
3      digit_b = reverse(parse.(Int, collect(b)))
4
5      m = length(digit_a)
6      n = length(digit_b)
7
8      result = zeros{Int64, m + n}
9
10     for i in 1:m
11         for j in 1:n
12             result[i + j - 1] += digit_a[i] * digit_b[j]
13         end
14     end
15
16     carry = 0
17     for k in 1:length(result)
18         result[k] += carry
19         carry = div(result[k], 10)
20         result[k] %= 10
21     end
22
23     while length(result) > 1 && result[end] == 0
24         pop!(result)
25     end
26
27     return join(reverse(result))
28 end
29
30 a = "1234"
31 b = "98765"
32 result = multiply(a, b)
33 println("Результат: ", result)

```

Быстрый столбик

```

• Activating new project at `C:\Users\nastd\.julia\environments\v1.11`
Результат: 121876010
* Terminal will be reused by tasks, press any key to close it.

```

Результат

## Выводы

Мы изучили 5 алгоритмов целочисленной арифметики многократной точности и реализовали их программно на языке программирования Julia.

## Список литературы

1. Mathematics // Julia URL: <https://docs.julialang.org/en/v1/base/math/>