

# Лабораторная работа №8: Целочисленная арифметика многократной ТОЧНОСТИ

Дисциплина: Математические основы защиты информации и информационной безопасности

---

Манаева Варвара Евгеньевна, НФИМд-01-24, 1132249514

21 декабря 2024

Российский университет дружбы народов, Москва, Россия

## Общая информация о лабораторной работе

---

Ознакомиться с целочисленной арифметикой многократной точности.

1. Реализовать алгоритмы из задания лабораторной работы.

## Теоретическое введение

---

В процессе лабораторной работы рассматриваются примеры работы с числами в  $b$ -ичной системе счисления,  $b \geq 2$  – натуральное число.

Числа для выполнения лабораторной работы будем записывать в виде:

$$u = u_1 u_2 u_3 \dots u_n.$$

## Выполнение лабораторной работы

---

# Алгоритм 1. Сложение неотрицательных целых чисел

## 1. Сложение неотрицательных целых чисел

$$12533 + 989 = 13522$$

```
[12]: function sum_accurate(u, v, b=10)
      k = 0
      u_str = parse.(Integer, only.(split(string(u), ""))); v_str = parse.(Integer, only.(split(string(v), "")))
      n_u = length(u_str); n_v = length(v_str)
      j = max(n_u, n_v)
      w = zeros(Int64, j+1)
      if n_u < n_v
          temp = zeros(Int64, j)
          temp[n_v - n_u + 1:j] = u_str
          u_str = [i for i in temp]
      elseif n_v < n_u
          temp = zeros(Int64, j)
          temp[n_u - n_v + 1:j] = v_str
          v_str = [i for i in temp]
      end
      while j != 0
          k_temp = (u_str[j] + v_str[j] + k) % b
          w[j+1] = k_temp
          k = round(Int, (u_str[j] + v_str[j] + k - k_temp) / b)
          j -= 1
      end
      w[1] = k
      return parse(Int, join(string.(w)))
end
```

```
[12]: sum_accurate (generic function with 2 methods)
```

```
[13]: sum_accurate(12533,989)
```

```
[13]: 13522
```



## Алгоритм 2. Вычитание неотрицательных целых чисел

### 2. Вычитание неотрицательных целых чисел

$$12533 - 989 = 11544$$

```
[14]: function raz_accurate(u, v, b=10)
    if u < v
        return string(u) * " should be greater than " * string(v)
    end
    k = 0
    u_str = parse.(Integer, only.(split(string(u), ""))); v_str = parse.(Integer, only.(split(string(v), "")))
    n_u = length(u_str); n_v = length(v_str)
    j = max(n_u, n_v)
    w = zeros{Int64, j}
    if n_v < n_u
        temp = zeros{Int64, j}
        temp[n_u - n_v + 1:j] = v_str
        v_str = [i for i in temp]
    end
    while j != 0
        if u_str[j] < v_str[j]
            k = b
            u_str[j-1] -= 1
        else
            k = 0
        end
        k_temp = (u_str[j] - v_str[j] + k) % b
        w[j] += k_temp
        j -= 1
    end
    return parse{Int, join(string.(w))}
end
```

```
[14]: raz_accurate (generic function with 2 methods)
```

```
[15]: raz_accurate(12533,989)
```

```
[15]: 11544
```

## Алгоритм 3. Умножение неотрицательных целых чисел

### 3. Умножение неотрицательных целых чисел столбиком

$$12533 * 989 = 12395137$$

```
[66]: function umn_accurate(u, v, b=10)
      k = 0
      u_str = parse.(Integer, only.(split(string(u), ""))); v_str = parse.(Integer, only.(split(string(v), "")))
      i = length(u_str); j = length(v_str)
      w = zeros{Int64, i + j}
      while j > 0
          i = length(u_str)
          k = 0
          while i > 0
              k_temp = u_str[i] * v_str[j] + w[i+j] + k
              w[i+j] = k_temp % b
              k = round{Int, (k_temp - w[i+j]) / b}
              i -= 1
          end
          w[j] = k
          j -= 1
      end
      return parse{Int, join(string.(w))}
end
```

```
[66]: umn_accurate (generic function with 2 methods)
```

```
[67]: umn_accurate(12533,989)
```

```
[67]: 12395137
```

## Алгоритм 4. Быстрое умножение столбиком

### 4. Быстрый столбик

$$12533 * 989 = 12395137$$

```
[78]: function umn_fast(u, v, b=10)
    u_str = parse(Integer, only.(split(string(u), ""))); v_str = parse(Integer, only.(split(string(v), "")))
    n = length(u_str); m = length(v_str)
    w = zeros(Int64, n + m)
    t = 0
    for s in 0:m+n-1
        for i in 0:s
            if n-i <= 0 || m-s+i <= 0
                continue
            end
            t += u_str[n-i] * v_str[m-s+i]
        end
        w[n+m-s] = t % b
        t = round(Int64, (t - w[n+m-s]) / b)
    end
    return parse(Int, join(string.(w)))
end
```

```
[78]: umn_fast (generic function with 2 methods)
```

```
[79]: umn_fast(12533,989)
```

```
[79]: 12395137
```

## Выводы

---

В результате работы мы ознакомились с целочисленной арифметикой многократной точности и реализовали 4 алгоритма.

Были записаны скринкасты:

- выполнения лабораторной работы;
- создания отчёта по результатам выполнения лабораторной работы;
- создания презентации по результатам выполнения лабораторной работы;
- защиты лабораторной работы.