Отчёт по лабораторной работе №8: Целочисленная арифметика многократной точности

Дисциплина: Математические основы защиты информации и информационной безопасности

Манаева Варвара Евгеньевна

Содержание

# 1 Общая информация о задании лабораторной работы

## 1.1 Цель работы

Ознакомиться с целочисленной арифметикой многократной точности.

## 1.2 Задание [1]

1. Реализовать алгоритмы из задания лабораторной работы.

# 2 Выполнение лабораторной работы [1]

## 2.1 Алгоритм 1. Сложение неотрицательных целых чисел

Исходный код написан на языке Julia [2]. Код функции, осуществляющей сложение неотрицательных целых чисел, представлен ниже.

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

### 2.1.1 Проверка работы функции

sum\_accurate(12533,989)

Результат работы кода представлен ниже (рис. 1).



Рис. 1: Сумма целых неотрицательных чисел

## 2.2 Алгоритм 2. Вычитание неотрицательных целых чисел

Исходный код написан на языке Julia [2]. Код функции, осуществляющей вычитание целых чисел, представлен ниже.

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

### 2.2.1 Проверка работы функции

raz\_accurate(12533,989)

Результат работы кода представлен ниже (рис. 2).

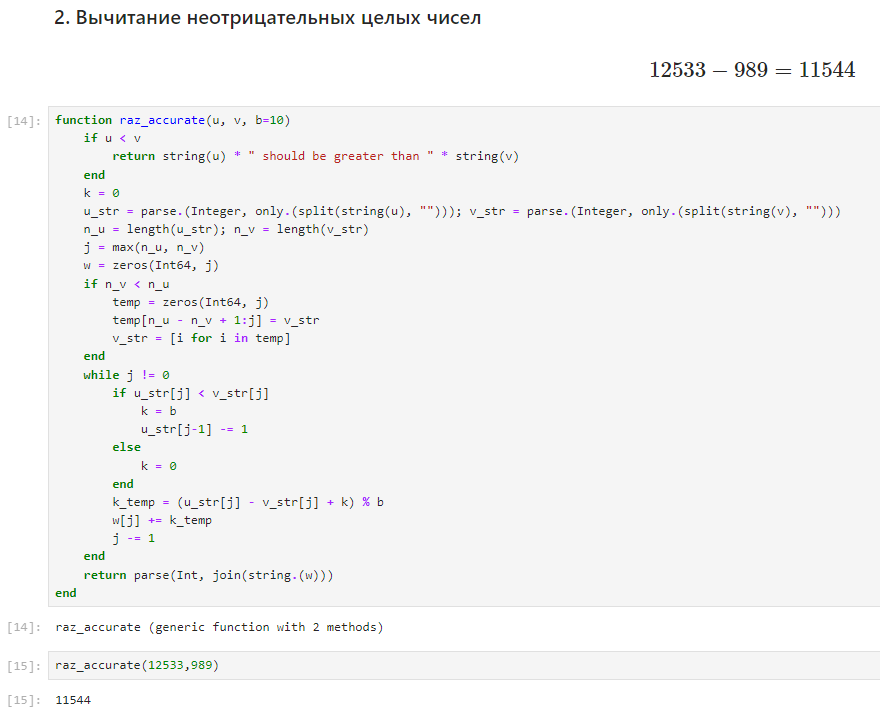


Рис. 2: Вычитание целых чисел

## 2.3 Алгоритм 3. Умножение неотрицательных целых чисел

Исходный код написан на языке Julia [2]. Код функции, осуществляющей умножение неотрицательных чисел, представлен ниже.

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

### 2.3.1 Проверка работы функции

umn\_accurate(12533,989)

Результат работы кода представлен ниже (рис. 3).

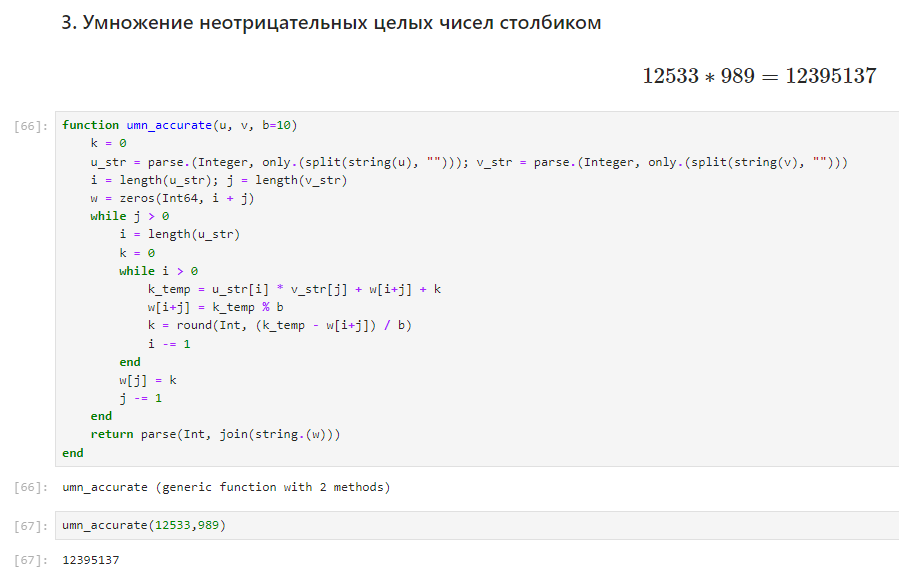


Рис. 3: Умножение неотрицательных чисел

## 2.4 Алгоритм 4. Быстрое умножение столбиком

Исходный код написан на языке Julia [2]. Код функции, осуществляющей быстрое умножение столбиком, представлен ниже.

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

### 2.4.1 Проверка работы функции

umn\_fast(12533,989)

Результат работы кода представлен ниже (рис. 4).

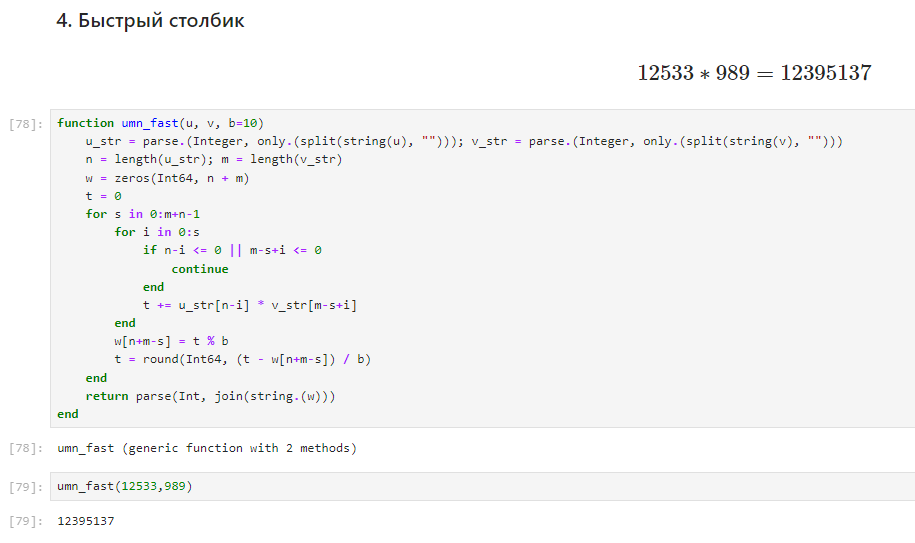


Рис. 4: Быстрое умножение столбиком

# 3 Выводы

В результате работы мы ознакомились с целочисленной арифметикой многократной точности.

Также были записаны скринкасты:

На RuTube:

* [Весь плейлист](https://rutube.ru/plst/540770)
* [Запись создания шаблона отчёта и презентации для заполнения](https://rutube.ru/video/f2eff0bf79aae34ebe62602bdb92a9b8)
* [Выполнения лабораторной работы](https://rutube.ru/video/702914c96130907b2c10014e01e35a83)
* [Запись создания отчёта](https://rutube.ru/video/078597536c1b0cd543c09ea6d9dff0e5/)
* [Запись создания презентации](https://rutube.ru/video/0301607c8f2f30b42639e7629a1ae2d4)
* [Защита лабораторной работы](https://rutube.ru/video/6d11d4bf50b3a1bbc63f0af210783a90/)

На Платформе:

* [Весь плейлист](https://plvideo.ru/playlist?list=vaNN02mO97J6)
* [Запись создания шаблона отчёта и презентации для заполнения](https://plvideo.ru/watch?v=xAma7VEEbvb-)
* [Выполнения лабораторной работы](https://plvideo.ru/watch?v=cmOkt1N2P1pk)
* [Запись создания отчёта](https://plvideo.ru/watch?v=glc_k6Jx9Bkg)
* [Запись создания презентации](https://plvideo.ru/watch?v=jSH3ZzgFrArk)
* [Защита лабораторной работы](https://plvideo.ru/watch?v=yVpnvSIMQIIv)

# Список литературы

1. Лабораторная работа №?. ?тема? [Электронный ресурс]. RUDN, 2024. URL: <https://esystem.rudn.ru/pluginfile.php/2368510/mod_folder/content/0/lab03.pdf>.

2. Julia 1.10 Documentation [Электронный ресурс]. 2024. URL: <https://docs.julialang.org/en/v1/>.