

Отчёт по лабораторной работе №8.

Целочисленная арифметика многократной точности

*Дисциплина: Математические основы защиты информации
и информационной безопасности*

Студент: Асеинова Елизавета

Группа: НФИмд-01-23

Преподаватель: д-р.ф.-м.н., проф. Кулябов Дмитрий Сергеевич

23 декабря, 2023, Москва

Цели и задачи работы

Целью данной лабораторной работы является ознакомление с алгоритмами по воплощению целочисленной арифметики многократной точности, а также программная реализация данных алгоритмов.

Реализовать рассмотренные в инструкции к лабораторной работе алгоритмы программно.

Алгоритмы:

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

Ход выполнения и результаты

Вспомогательные действия

```
import math

str2num = {chr(letter_ord) : (letter_ord - ord("A") + 10) for letter_ord in range(ord("A"), ord("Z") + 1)}
for num in "0123456789":
    str2num[num] = int(num)
num2str = {value: key for (key, value) in str2num.items()}

def add_0(u, n, array = False):
    result = [0] * (n - len(u))
    if array:
        result.extend(u)
    return result
return "".join([str(i) for i in result]) + u
```

Рис. 1: Вспомогательные действия для удобства дальнейших вычислений

Алгоритм 1. Сложение неотрицательных целых чисел.

Реализация

✓ Алгоритм 1

```
def alg_1(u_s, v_s, b):  
    u = [str2num[letter] for letter in u_s]  
    v = [str2num[letter] for letter in v_s]  
    if len(u) != len(v):  
        if len(u) < len(v):  
            u = add_0(u, len(v), True)  
        else:  
            v = add_0(v, len(u), True)  
    n = len(u)  
    k = 0  
    w = []  
    for j in range(n-1, -1, -1):  
        w.append((u[j] + v[j] + k) % b)  
        k = math.floor((u[j] + v[j] + k) / b)  
    w.append(k)  
    w.reverse()  
    return "".join([num2str[num] for num in w])
```

```
[10] print(alg_1("221", "1567", 10))  
      print(alg_1("807", "M16", 10))
```

```
01788  
3323
```

Рис. 2: Алгоритм 1. Сложение неотрицательных целых чисел

Алгоритм 2. Вычитание неотрицательных целых чисел.

Реализация

✓ Алгоритм 2

```
✓ [11] def alg_2(u_s, v_s, b):  
0     u = [str2num[letter] for letter in u_s]  
JMK   v = [str2num[letter] for letter in v_s]  
      if len(u) != len(v):  
        if len(u) < len(v):  
          u = add_0(u, len(v), True)  
        else:  
          v = add_0(v, len(u), True)  
      elif u < v:  
        return "u должно быть больше v"  
      n = len(u)  
      k = 0  
      w = []  
      for j in range(n-1, -1, -1):  
        w.append((u[j] - v[j] + k) % b)  
        k = math.floor((u[j] - v[j] + k) / b)  
      w.append(k)  
      w.reverse()  
      return "".join([num2str[num] for num in w])
```

```
✓ [13] alg_2("715", "215", 10)  
0  
JMK
```


Алгоритм 3. Умножение неотрицательных целых чисел столбиком. Реализация

✓ Алгоритм 3

```
✓ [14] def alg_3(u_s, v_s, b):  
0   u = [str2num[letter] for letter in u_s]  
cek v = [str2num[letter] for letter in v_s]  
    n = len(u)  
    m = len(v)  
    w = [0] * (m+n)  
    for j in range(m-1, -1, -1):  
        if v[j] != 0:  
            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  
    return "".join([num2str[num] for num in w])
```

```
✓ [16] alg_3("650", "1000", 10)  
0  
cek  
      '0650000'
```

Алгоритм 4. Быстрый столбик. Реализация

✓ Алгоритм 4

```
0  def alg_4(u_s, v_s, b):  
JSC  
    u = [str2num[letter] for letter in u_s]  
    v = [str2num[letter] for letter in v_s]  
    n = len(u)  
    m = len(v)  
    w = [0] * (m+n)  
    t = 0  
    for s in range(0, m+n):  
        for i in range(0, s+1):  
            if (0 <= n-i-1 < n) and (0 <= m-s+i < m):  
                t = t + u[n-i-1] * v[m-s+i-1]  
            w[m+n-s-1] = t % b  
            t = math.floor(t/b)  
    return "".join([num2str[num] for num in w])
```

```
0  [18] alg_4("650", "1000", 10)  
JSC  
      '0650000'
```

Рис. 5: Алгоритм 4. Быстрый столбик

Алгоритм 5. Деление многоразрядных целых чисел. Реализация

▼ Алгоритм 5

```
def to10(u_str, b, array = False):  
    u_array = u_str if array else [str2num[letter] for letter in u_str]  
    u = 0  
    for i in range(len(u_array)):  
        u += (b ** i) * u_array[len(u_array) - i - 1]  
    return u  
  
def to_b(number, b, n = 1):  
    (q, r) = (math.floor(number / b), number % b)  
    w = num2str[r]  
  
    while q >= b:  
        (q, r) = (math.floor(q / b), q % b)  
        w = w + num2str[r]  
  
    if q != 0: w = w + num2str[q]  
  
    while len(w) < n:  
        w = w + "0"  
  
    return w[::-1]  
  
def trim_zero(a):  
    while a[0] == '0' and len(a) > 1:  
        a = a[1:]  
    return a  
  
def algorithm_5(u_str, v_str, b):  
    u = u_str  
    v = v_str  
    u_10 = to10(u, b)  
    v_10 = to10(v, b)  
    n = len(u) - 1  
    t = len(v) - 1  
  
    if v[0] == 0 or not (n >= t >= 1):  
        return "Некорректные входные данные"  
  
    q = [0] * (n - t + 1) # шаг 1  
  
    while u_10 >= v_10 * (b ** (n - t)): #  
        q[n - t] = q[n - t] + 1  
        u_10 = u_10 - v_10 * (b ** (n - t))
```

Алгоритм 5. Деление многоразрядных целых чисел. Реализация

[illegible]

Рис. 7: Алгоритм 5. Деление многоразрядных целых чисел

В результате выполнения данной лабораторной работы нам удалось осуществить программно алгоритмы, рассмотренные в описании к лабораторной работе.