Лабораторная работа №8

Целочисленная арифметика многократной точности

Кубасов В.Ю.

Содержание

# 1 Цель работы

Реализовать алгоритмы стандартных математических операций для многоразрядных чисел

# 2 Задание

Реализовать алгоритмы:  
- сложения  
- вычитания  
- умножения  
- деления

# 3 Выполнение лабораторной работы

## 3.1 Сложение

def add(u = "", w = "", b = 10):  
 k = 0  
 ans = ""  
   
 for i in range(len(u))[::-1]:  
 ans = chr((ord(u[i]) + ord(w[i]) - 2 \* ord('0') + k) % b + ord('0')) + ans  
 k = int((ord(u[i]) + ord(w[i]) - 2 \* ord('0') + k) / b)  
   
 if (k > 0):  
 ans = '1' + ans  
 print(ans)  
   
add("9824211231231232131", "9824211231231232131")

## 3.2 Вычитание

def sub(u = "", w = "", b = 10):  
 k = 0  
 ans = ""  
   
 for i in range(len(u))[::-1]:  
 ans = chr(abs(ord(u[i]) - ord(w[i]) + k) % b + ord('0')) + ans  
 k = int((ord(u[i]) - ord(w[i]) + k) / b)  
   
 print(ans)  
   
sub("9824211231231232131", "9824211231231232131")

## 3.3 Умножение

ZERO = ord('0')  
  
def mul(u = "", v = "", b = 10):  
 k = 0  
 ans = [0] \* (len(u) + len(v))  
   
 for j in range(len(v))[::-1]:  
 if (v[j] == 0):  
 ans[j] = 0  
 else:  
 k = 0  
 for i in range(len(u))[::-1]:  
 t = (ord(v[j]) - ZERO) \* (ord(u[i]) - ZERO) - + ans[i + j] + k  
 ans[i + j] = t % b  
 k = int(t / b)  
 ans[j] = k   
   
 print(''.join(str(x) for x in ans))  
   
mul("9824211231231232131", "9824211231231232131")

## 3.4 Умножение быстрое

def mulf(u = "", v = "", b = 10):  
 t = 0  
 ans = [0] \* (len(u) + len(v))  
  
 for s in range(len(ans) - 1):  
 for i in range(s):  
 t = (ord(u[len(u) - 1 - i]) - ord('0')) \* (ord(v[len(v) - 1 - s + i]) - ord('0'))  
 ans[len(v) + len(u) - s] = t % b  
 t = int(t / b)  
   
 print(''.join(str(x) for x in ans))  
   
mulf("9824211231231232131", "9824211231231232131")

## 3.5 Деление с утечкой памяти

ZERO = ord('0')  
  
def div(u = "", v = "", b = 10):  
 t = 0  
 q = [0] \* (len(u) - len(v))  
 r = [0] \* len(v)  
   
 while(u >= v \* b \*\* (len(u) - len(v))):  
 for i in range(len(v) + 1, len(u) + 1)[::-1]:  
 if (u[i] > v[0]):  
 q[i - len(v) - 1] = b - 1  
 else:  
 q[i - len(v) - 1] = int((u[i] \* b + u[i - 1]) / v[0])  
   
 while(q[i - len(v) - 1] \* (v[0]\*b + v[1]) > u[i] \* b\*\*2 + u[i - 1] \* b + u[i - 2]):  
 --q[i - len(v) - 1]  
   
 u = u - q[i - len(v) - 1] \* b \*\* (i - len(v) - 1) \* v  
   
 if (u < 0):  
 u = u + v \* b\*\*(i - len(v) - 1)  
 --q[i - len(v) - 1]  
   
 r = u  
   
 print('\n' + ''.join(str(x) for x in ans))  
 print('\n' + ''.join(str(x) for x in r))  
   
div("9824125675631232131", "24142", 10)

# 4 Выводы:

* В ходе работы реализовали предложенные алгоритмы

# Список литературы