



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА

ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)

О Т Ч Е Т

По лабораторной работе № 1

Название: "Длинная арифметика": обработка больших чисел

Дисциплина: Типы и структуры данных

Студент

ИУ7-31Б

(Группа)

Савинова М. Г.

(Фамилия И. О.)

Преподаватель

Барышникова М. Ю.

Москва, 2022

Оглавление

Отчет.....	1
"Длинная арифметика": обработка больших чисел.....	1
Условие задачи.....	2
Описание ТЗ.....	3
1. Описание входных данных.....	3
2. Описание выходных данных.....	4
3. Способ обращения к программе.....	5
4. Описание возможных аварийных ситуаций и ошибок пользователя.....	5
Описание внутренних структур данных.....	5
Описание алгоритма.....	6
Тестовые данные.....	7
1. Позитивные тесты.....	7
2. Негативные тесты.....	8
Выводы.....	9
*Ответы на контрольные вопросы.....	9

Условие задачи

Смоделировать операцию умножения целое число $\pm X$ длиной до 30 десятичных цифр, на действительное числа в форме $\pm m.nE \pm k$, где суммарная длина мантииссы ($m+n$) - до 30 значащих цифр, а величина порядка К - до 5 цифр. Результат выдать в форме $\pm 0.m1E \pm k1$, где $m1$ - до 30 значащих цифр, а $k1$ - до 5 цифр.

Описание ТЗ

1. Описание входных данных

Сначала на вход программе поступает строка, содержащая целое число в виде:

$\pm X$

Ограничения:

- максимальная длина числа — **30 цифр**
- если знак в начале числа не указан, то по умолчанию число **положительное**
- если был введен 0, то результат умножения будет «+», вне зависимости от знака 2го числа
- число считается некорректно заданным, если в начале числа идут «0»
- число считается некорректно заданным, если в числе присутствуют символы, отличные от цифр

Тесты:

Корректный ввод	Некорректный ввод
1	++1
+1	1-
-1	1.0
1000000000000000000000000000000(30 символов)	10000000000000000000000000000000(31 символ)
	a1a
	00000000

символов в мантиссе)	в мантиссе)
	пустая строка

2. Описание выходных данных

В результате выполнения программы будет выведено вещественное число в виде:

$$\pm 0.m1E\pm k1$$

Внешние спецификации:

- длина мантиссы m1- до 30 цифр
- максимальный порядок - 5 символов
- в случае превышения длины мантиссы, происходит округление: если 31 цифра ≥ 5 , то значение 30 цифры увеличивается на 1, иначе 31 цифра игнорируется
- в случае превышения порядка, на экран выводится соответствующее сообщение

3. Способ обращения к программе

Обращение к программе происходит через консоль, путём запуска скрипта **run.sh**, который сначала производит релизную сборку проекта, а затем запускает его.

4. Описание возможных аварийных ситуаций и ошибок пользователя

Аварийная ситуация	Код завершения	Сообщение
В целом числе недопустимые символы	1	<i>Invalid input for integer!</i>
Длина целого числа больше 30	2	<i>Invalid length for integer!</i>
В вещественном числе недопустимые символы	3	<i>Invalid input for float!</i>
Длина вещественного числа больше 30	4	<i>Invalid length for float!</i>
Порядок вещественного числа вне допустимого значения	5	<i>Invalid power for float!</i>
Порядок вещественного числа в результате умножения вне допустимого значения	6	<i>Stack overflow!</i>

Описание внутренних структур данных

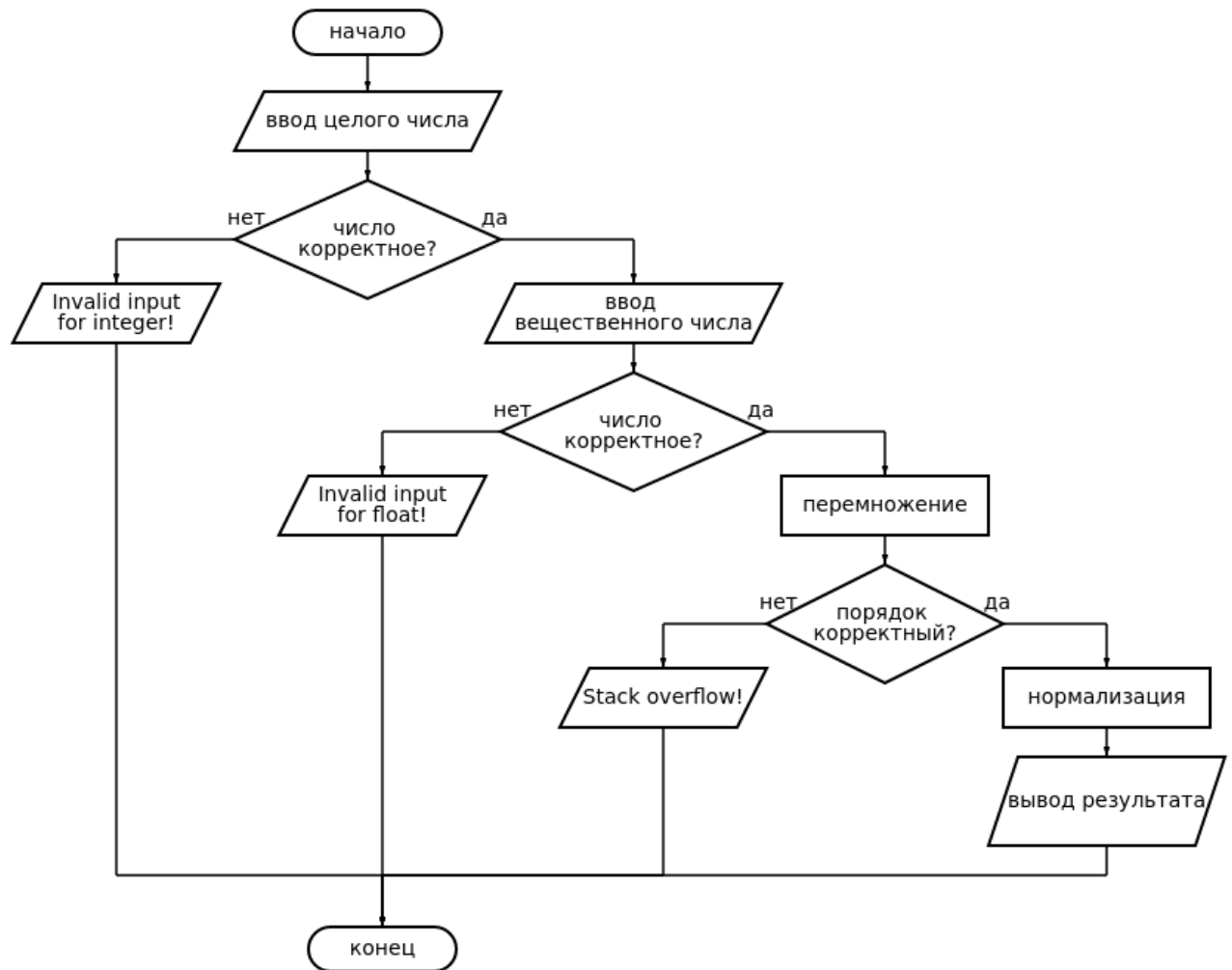
- «Большое» вещественное число - ***float_num***

```
typedef struct
{
    short int sign;           // знак действительного числа
    int len;                  // длина мантииссы
    int power;                // значение порядка
    int numbers[MAX_MANTISSA]; // массив, для хранения мантииссы
} float_num;
```

- «Большое» целое число - ***int_num***

```
typedef struct
{
    short int sign;           // знак целого числа
    int len;                  // длина целого числа
    int power;                // значение порядка
    int numbers[MAX_MANTISSA]; // массив, для хранения целого числа
} int_num;
```

Описание алгоритма



Тестовые данные

1. Позитивные тесты

1. Целое число «0»
2. Вещественное число «0»
3. В целом числе 30 цифр
4. В результате максимальное значение порядка
5. В результате минимальное значение порядка
6. Вещественное число задается с «.»
7. Вещественное число задается с «,»
8. Вещественное число задается как целое

9. Целое число отрицательное; вещественное положительное
10. Целое число положительно; вещественное отрицательное
11. Целое и вещественное число отрицательные
12. Результат округления в меньшую сторону
13. Результат округления в большую сторону
14. Нормализация

[illegible]

2. Негативные тесты

1. В целом числе больше 30 цифр
2. В вещественном числе больше 30 цифр
3. Порядок вещественного числа больше 5 цифр
4. В целом числе несколько незначащих 0
5. В вещественном числе несколько «0» до точки
6. В вещественном числе несколько «0» после точки
7. В порядке вещественного числа несколько незначащих «0»

8. В целом числе недопустимые символы
9. В вещественном числе недопустимые символы
10. В порядке вещественного числа недопустимые символы
11. Целое число - пустая строка
12. Вещественное число - пустая строка
13. Значение порядка меньше допустимого значения
14. Значение порядка больше допустимого значения

[illegible]

Выводы

В данной лабораторной работе было изучено представление чисел в памяти компьютера. А также выяснили, что для работы с большими числами, превышающими размер регистра, программисту необходимо самостоятельно конструировать тип данных и методы для обработки «больших чисел».

***Ответы на контрольные вопросы**

1. Каков возможный диапазон чисел, представляемых в ПК?

Каждый тип данных определяет диапазон значений, который в свою очередь зависит от размера области памяти, выделяемой для хранения переменной этого типа, от знака в числе, от типа представления: целое или вещественное.

К примеру, если под хранение целого положительного числа выделено 16 разрядов, то его максимальное значение не может превышать $2^{16}-1=65\,535$, а если целое может принимать как положительные, так и отрицательные, то: $[-2^{15}; 2^{15}-1]$

2. Какова возможная точность представления чисел, чем она определяется?

Точность представления вещественного числа зависит от максимально возможной длины мантиссы, которая, опять-таки, зависит от области выделяемой памяти и наличия знака. Если длина мантиссы выходит за границы разрядной сетки, то происходит округление. Чем больше разрядов отводится под мантиссу, тем выше точность представления числа.

Максимальная длина мантиссы — 52 разряда, а порядка — 11.

3. Какие стандартные операции возможны над числами?

Сложение, вычитание, умножение, деление, сравнение(только с целыми числами)

4. Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?

Для представления положительного целого числа можно использовать массив, в каждой ячейке которого размещена 1 цифра. В случае более «сложного» типа данных, можно воспользоваться структурой, которая также будет хранить массив цифр мантиссы, массив порядка, знак порядка и мантиссы.

5. Как можно осуществить операции над числами, выходящими за рамки машинного представления?

Можно занести эти числа в массив, и с конца выполнять операции с их соответствующими элементами.