

## Оглавление

<b>Оглавление</b>	<b>2</b>
Описание условия задачи	3
Описание технического задания	3
Входные данные:	3
Выходные данные:	3
Действие программы:	3
Обращение к программе:	4
Описание структуры данных	5
Описание алгоритма	5
Набор тестов	6
Ответы на контрольные вопросы	8
1. Каков возможный диапазон чисел, представляемых в ПК?	8
2. Какова возможная точность представления чисел, чем она определяется?	8
3. Какие стандартные операции возможны над числами?	8
4. Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?	8
5. Как можно осуществить операции над числами, выходящими за рамки машинного представления?	9
<b>ВЫВОД</b>	<b>9</b>

## Описание условия задачи

Смоделировать операцию деления целого числа длиной до 40 десятичных цифр на действительное число в форме  $\pm m.n \text{ E } K$ , где суммарная длина мантиссы ( $m+n$ ) - до 40 значащих цифр, а величина порядка  $K$  - до 5 цифр. Результат выдать в форме  $\pm 0.m1 \text{ E } \pm K1$ , где  $m1$  - до 40 значащих цифр, а  $K1$  - до 5 цифр.

## Описание технического задания

### Входные данные:

**Целое число:** строка, содержащая целое число в виде  $[+\backslash-m]$ . Если не указать знак перед числом – по умолчанию будет считаться за '+'. Длина числа  $m$  - 40 цифр.

**Действительное число:** строка, содержащая вещественное число в форме  $\pm m.n[E/e]\pm K$ . Если не указать знак перед числом и/или экспонентой – по умолчанию будет считаться за '+'. Максимальная длина мантиссы — 40 цифр, максимальная длина порядка — 5 цифр.

### Выходные данные:

Строка в виде  $[+-]0.me[+/-]K1$ . Длина мантиссы — 40 цифр; длина порядка — до 5 цифр.

### Действие программы:

Деление целого числа на действительное.

## **Обращение к программе:**

Программа запускается через терминал, в командной строке, с помощью команды `./app.exe`. После запуска пользователь увидит приглашение к вводу. Далее вводятся числа. Если запускать программу с флагом `-h`, то на экран выводится справочная информация.

### **Аварийные ситуации:**

1. Некорректный ввод: ввод символа при вводе целого числа

Код ошибки - 1.

2. Некорректный ввод: ввод лишних символов при вводе вещественного числа

Код ошибки - 1.

3. Некорректный ввод: превышение длины целого числа

Код ошибки - 2.

4. Некорректный ввод: превышение длины мантиссы при вводе вещественного числа

Код ошибки - 2.

5. Некорректный ввод: превышение порядка при вводе вещественного числа

Код ошибки - 3.

6. Некорректный ввод: переполнение строки при вводе целого числа

Код ошибки - 4.

7. Некорректный ввод: переполнение строки при вводе вещественного числа

Код ошибки - 4.

8. Некорректный ввод: Введенное число не является целым

Код ошибки - 7.

9. Некорректный ввод: Пустой ввод

Код ошибки - 9.

10. Поведение программы: Деление на ноль

Код ошибки - 10.

Во всех нештатных ситуациях, программа выводит место, где произошла ошибка.

## Описание структуры данных

И целое, и вещественное, хранятся в структуре long\_number.

```
typedef struct
{
    unsigned short sign; // 0 - минус, 1 - плюс
    unsigned short mantise_size;
    unsigned short order;
    short mantise[MANTISE_DEFINE];
} long_number;
```

Листинг 1. Структура long\_number

Поля структуры:

1. sign - знак числа хранится в переменной;
2. mantise\_size - размер мантииссы
3. order - порядок числа
4. mantise - мантиисса числа.

## Описание алгоритма

1. Программа считывает две строки, первая содержит целое число, а вторая действительное.
2. Нормализация ввода.
3. Если действительное число не равно нулю - делим.
4. Ищется неполное делимое
5. В цикле ищется частное неполного делимого и делителя, остаток записывается в неполное делимое.
6. Нормализация числа.

7. Вывод результата на экран.

## Описание основных функций

```
int input_string(char *string)
```

Функция реализует ввод строки из stdin.

**Входные параметры:** указатель на строку.

**Выходные параметры:** измененная строка.

**Возвращаемый результат:** код возврата.

```
int input_int_number(long_number *number)
```

Функция реализует ввод целого числа

**Входные параметры:** указатель на структуру с длинным числом.

**Выходные параметры:** значение, записанное в структуру с длинным числом.

**Возвращаемый результат:** код возврата.

```
int input_real_number(long_number *number)
```

Функция реализует ввод действительного числа

**Входные параметры:** указатель на структуру с длинным числом.

**Выходные параметры:** значение, записанное в структуру с длинным числом.

**Возвращаемый результат:** код возврата.

```
void long_number_normalization(long_number *number)
```

Функция нормализует действительное число

**Входные параметры:** указатель на структуру с длинным числом.

**Выходные параметры:** измененная структура с длинным числом.

```
void find_part_divisible(long_number *part_divisible, long_number divisible, long_number divider)
```

Функция поиска неполного делимого

**Входные параметры:** указатель на неполное делимое, структура с делимым, структура с делителем.

**Выходные параметры:** измененная структура с неполным делителем.

```
int long_divisible(long_number divisible, long_number divider, long_number *result)
```

Функция деления длинного числа на длинное число.

**Входные параметры:** делимое, делитель, указатель на частное.

**Выходные параметры:** частное деления, записанное в структуру.

**Возвращаемый результат:** код возврата.

```
void rounding(long_number *result, int last_number)
```

Функция реализует округление длинного числа

**Входные параметры:** указатель на структуру с результатом, последняя цифра мантиссы.

**Выходные параметры:** измененная структура с результатом.

## Набор тестов

№	Описание теста	Первое число	Второе число	Вывод
1	Деление единицы на единицу (в обычном виде)	1	1	0.1e1
2	Деление единицы на единицу в экспоненциальной форме	1	0.1e1	0.1e1
3	Деление на меньшее число	1	0.5	0.2e1
4	Деление меньшего на большее	1	2	0.5e0
5	Деление с уменьшением порядка	1	10	0.1e0
6	Деление с изменением знака	1	-100	-0.1e-1

7	Деление двух отрицательных чисел	-1	-2	0.5e0
8	Деление граничных значений	1111111111 1111111111 1111111111 111111	21	0.5291005 291005291 005291005 291005291 005285e37
9	Деление длинного целого на действительное	2222222222 2222222222 2222222222 2222222	1.11111111 1111111111 1111111111 11111	2e57
10	Деление на ноль	12	0	Error, division on zero
11	Переполнение строки	1111111111 1111111111 1111111111 1111111111 1111111111 1111111111		Error, overflow input
12	Символ в записи числа	1a	2.121a	Error, number must not contain any characters
13	Целое число записано с вещественной частью	1.12		Error, number must not contain any characters
14	Целое число записано с экспонентой	1e10		Error, number must not contain any characters

15	Вещественное число записано с лишними символами		+123.121e+10.	Error, number must not contain any characters
16	Ошибка с порядком числа		1e4444444	Error in order size. It's must be short then 5 digits
17	Пустой ввод			Error, input is empty
18	Деление с округлением	2	3	0.6666666 66666667

## Ответы на контрольные вопросы

### 1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от разрядности процессора и типа переменной. Максимальное значение 64-разрядного беззнакового целого числа (unsigned long long int) равно 18 446 744 073 709 551 615.

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

Точность представления вещественных чисел определяется количеством памяти, выделяемой для хранения мантиссы числа. Для мантиссы числа типа double выделяется 52 бита, с помощью этого мантисса числа может иметь значение до 4 503 599 627 370 496.



### **3. Какие стандартные операции возможны над числами?**

Операции сложения, вычитания, умножения, деление, взятие остатка, сравнения.

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

Программист может создать собственную структуру, где можно записать мантиссу, знак числа и порядка. Также может использовать массив символов.

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

Нужно использовать самостоятельно разработанные функции. Те, которые поразрядно умножают/складывают/делят/возводят в степень/сравнивают числа.

## **ВЫВОД**

Разработанный тип данных является очень важной структурой данных потому, что его длина ограничена только размером памяти компьютера/аппаратного стека. С такими числами можно проводить физические расчеты с очень большой точностью или хранить очень большие числа. Такой тип данных можно использовать, например, в астрофизике.