

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №3

по дисциплине: Основы программирования

тема: «Побитовые операции в Си»

Выполнил: ст. группы ПВ-201
Машуров Дмитрий Русланович

Проверил:
Притчин Иван Сергеевич
Брусенцева Валентина
Станиславовна

Белгород 2021 г.

Лабораторная работа № 4

«Преобразование типов»

Цель работы: получение навыков преобразования последовательности символов в числовое значение и наоборот.

Задания для подготовки к работе:

1. Изучить, в каких случаях и по каким правилам в языке Си преобразования типов выполняются автоматически.
2. Изучить возможности для явного преобразования типов в языке Си.
3. Разработать алгоритм и составить программу для решения задачи соответствующего варианта. Необходимые преобразования описать функциями. Библиотечные функции для ввода и вывода числовых значений и функции преобразования типов не использовать.
4. Подобрать наборы тестовых данных.

Задание варианта №17

С клавиатуры вводится последовательность вещественных чисел, числа разделены запятыми. Конец ввода – конец файла. Записать введенную последовательность в строку в форме с фиксированной точкой, сохранив три знака после точки.

1. Описание алгоритма и выделение подзадач

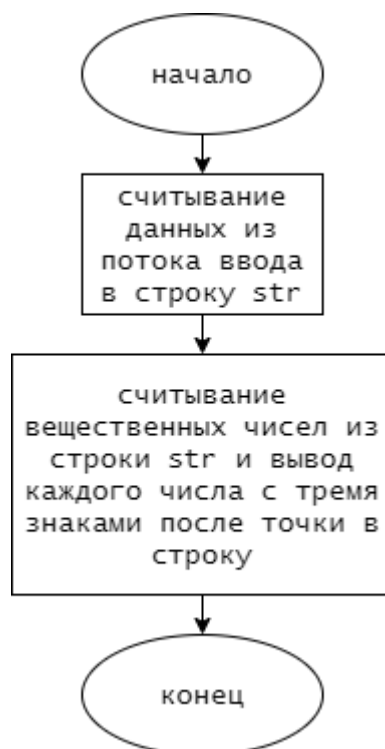
Будем считывать введенную строку в переменную, затем считывать числа.

Мы будем считывать число целиком и запоминать количество знаков k после точки. Далее считанное число будем делить на 10^k , чтобы перенести запятую на необходимое количество знаков

Выделим следующие подзадачи:

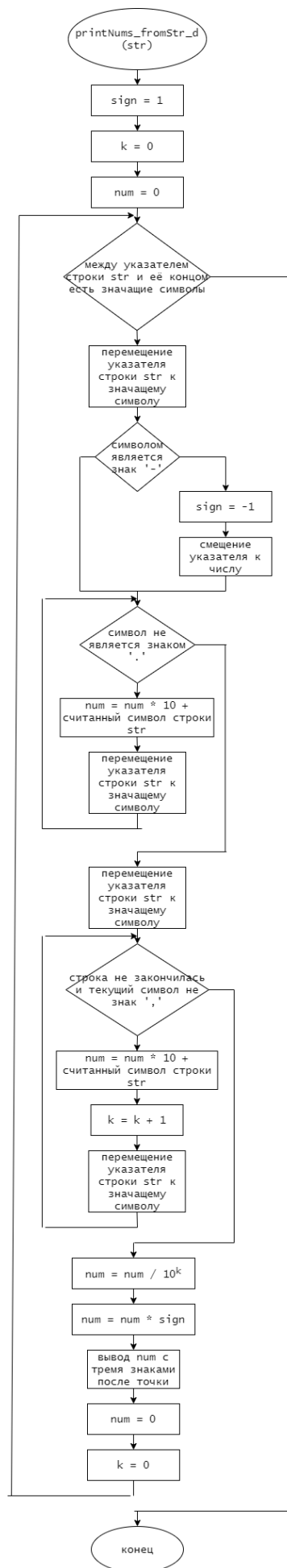
- 1) Считывание вещественных чисел из строки и вывод их с тремя знаками после точки

2. Блок-схема с укрупнёнными блоками



3. Описание подпрограмм

- 1) Считывание вещественных чисел из строки и вывод их с тремя знаками после точки
 - a) Заголовок: `void printNums_fromStr_d(char *str)`
 - b) Назначение: считывает вещественные числа из строки `str` и выводит их с тремя знаками после точки
- Блок-схема:



4. Тестовые данные

| № | Вход | Выход |
|---|------------------------------|-------------------------------|
| 1 | «123.2314, 343.22, 1.1» | 123.231 343.22 1.100 |
| 2 | «-8.56, -3333.3, -123.43123» | -8.560 -3333.300 - 123.431 |

5. Текст программы

```
/*
 * С клавиатуры вводится последовательность
 * вещественных чисел, числа разделены запятыми.
 * Конец ввода – конец файла.
 * Записать введенную последовательность в строку в форме
 * с фиксированной точкой, сохранив три знака после точки
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// считывает вещественные числа из строки str и
// выводит их с тремя знаками после точки.
void printNums_fromStr_d(char *str)
{
    int sign = 1;
    size_t k = 0;
    double num = 0;

    while (*str != '\0')
    {
        while (*str == ' ' || *str == ',') str++;

        if (*str == '-')
        {
            sign = -1;
            str++;
        }

        while (*str != '.')
        {
            num = num * 10 + (*str - '0');
            str++;
        }

        str++;

        while (*str != ',' && *str != '\0')
        {
            num = num * 10 + (*str - '0');
            k++;
            str++;
        }

        num /= (pow(10,k));

        num *= sign;

        printf("%.3lf ", num);
    }
}
```

```
        num = 0; k = 0;
    }
}

int main()
{
    char str[255];
    gets(str);
    printNums_fromStr_d(str);
}
```

6. Результаты работы и скрины программы

Скрины программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
// считывает вещественные числа из строки str и
// выводит их с тремя знаками после точки.
void printNums_fromStr_d(char *str)
{
    int sign = 1;
    size_t k = 0;
    double num = 0;

    while (*str != '\0')
    {
        while (*str == ' ' || *str == ',') str++;

        if (*str == '-')
        {
            sign = -1;
            str++;
        }

        while (*str != '.')
        {
            num = num * 10 + (*str - '0');
            str++;
        }

        str++;

        while (*str != ',' && *str != '\0')
        {
            num = num * 10 + (*str - '0');
            k++;
            str++;
        }
    }
}
```



```

        num /= (pow(10,k));

        num *= sign;

        printf("%.3lf ", num);

        num = 0; k = 0;
    }
}

int main()
{
    char str[255];
    gets(str);
    printNums_fromStr_d(str);
}

```

Результаты работы:

Пример №1:

```

123.2314, 343.22, 1.1
123.231 343.220 1.100

```

Пример №2:

```

-8.56, -3333.3, -123.43123
-8.560 -3333.300 -123.431

```

7. Анализ допущенных ошибок

- нет