

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 №4**

по дисциплине: Основы программирования  
тема: «Преобразование типов»

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**

Написать 4 функции преобразования данных:

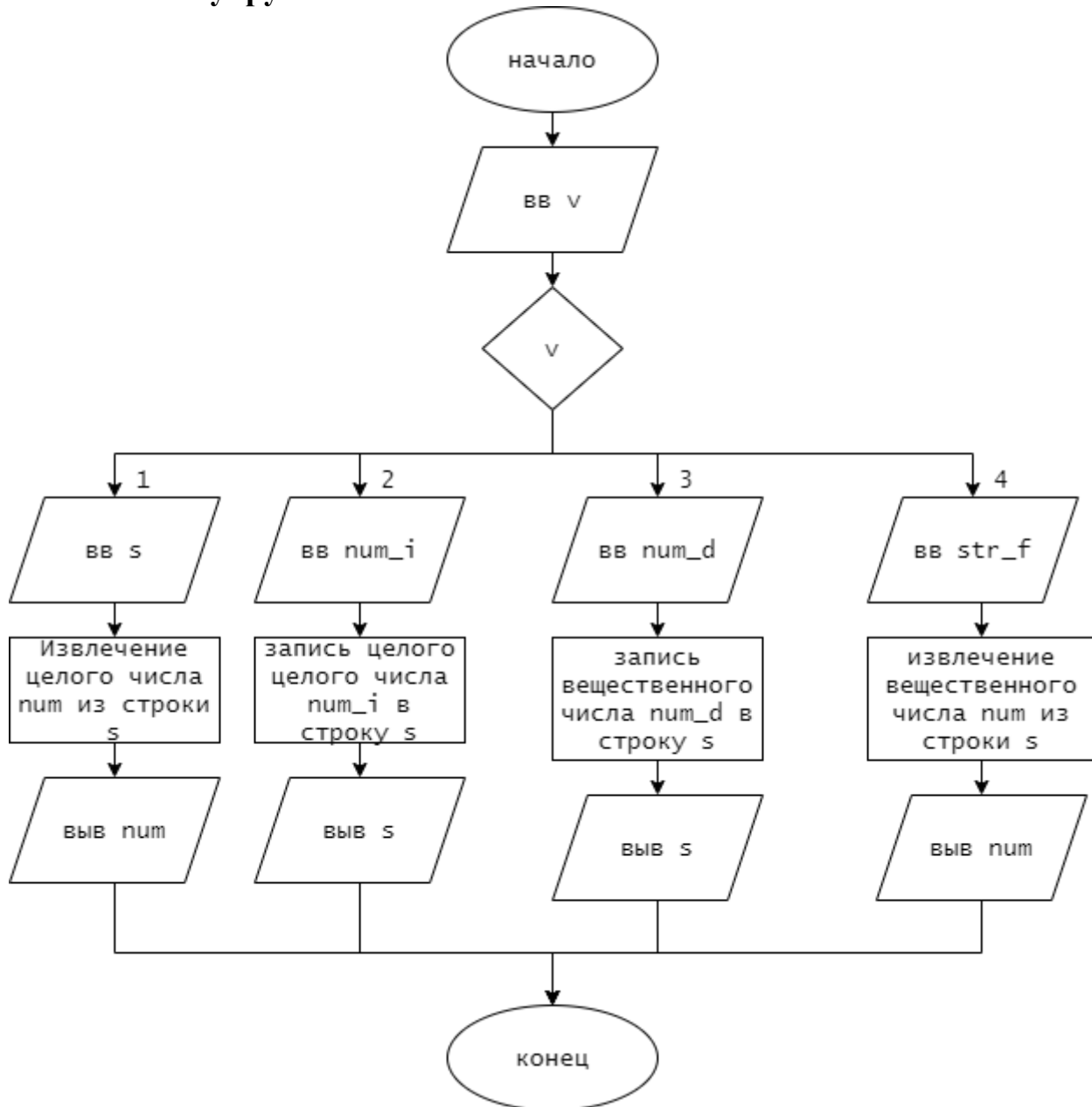
- 1) Извлечение целочисленного числа из строки
- 2) Извлечение вещественного числа из строки
- 3) Запись целочисленного числа в строку
- 4) Запись вещественного числа в строку

## 1. Выделение подзадач

Выделим следующие подзадачи:

- 1) Извлечение целого числа из строки
- 2) Извлечение вещественного числа из строки
  - а. Возведение 10 в определённой степени
- 3) Запись вещественного числа в строку
  - а. Получение нормальной формы числа
  - б. Конкатенация строк
- 4) Запись целого числа в строку
  - а. Переворот строки
    - і. Нахождение длины строки

## 2. Блок-схема с укрупнёнными блоками



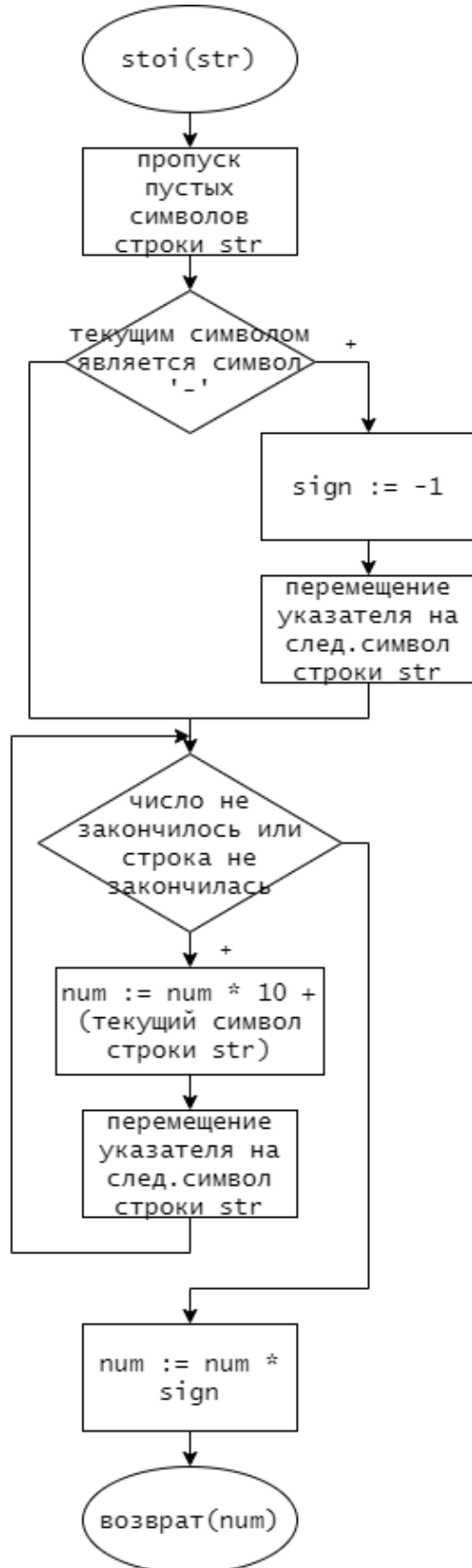
### 3. Описание подпрограмм

#### 1) Извлечение целого числа из строки

a) Заголовок: `int stoi(const char *str)`

b) Назначение: возвращает целое число, извлечённое из строки `str`

Блок-схема:



2) Извлечение вещественного числа из строки

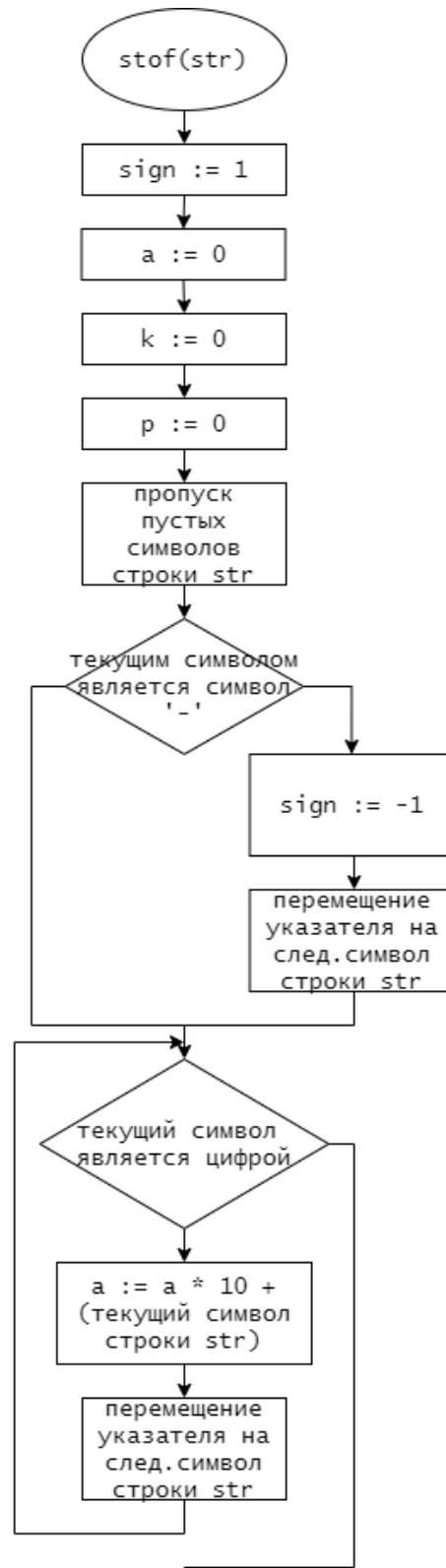
a) Выделение подзадачи:

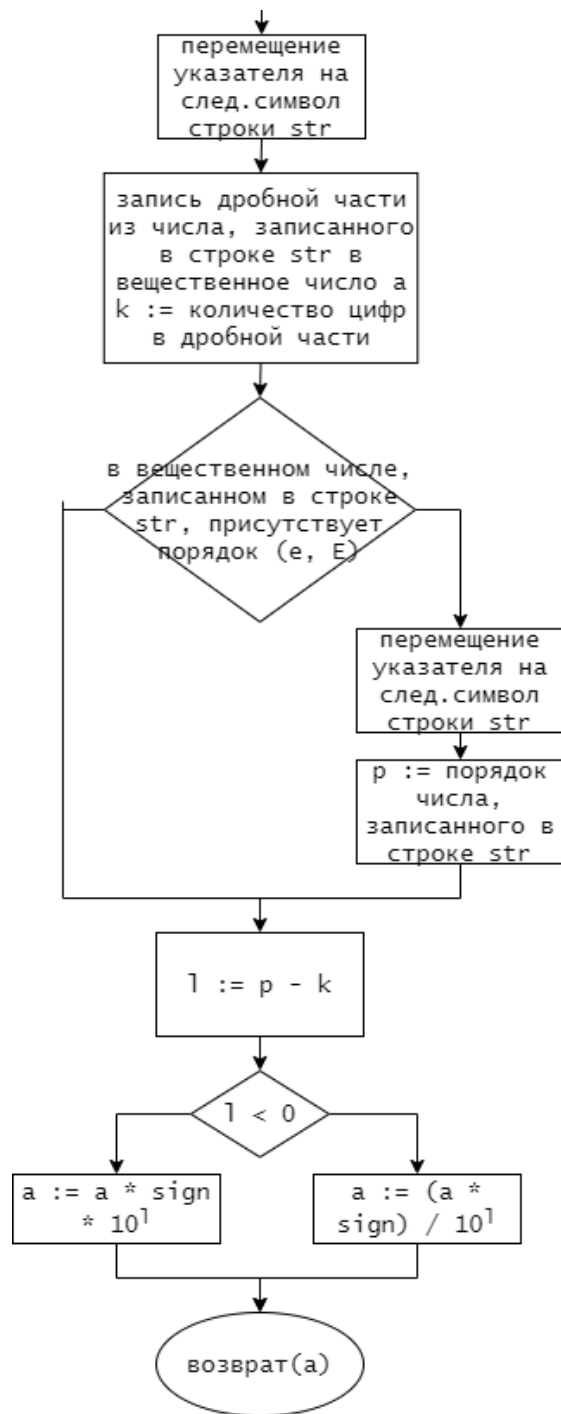
Возврат 10 в определённой степени

b) Заголовок: `double stof(const char *str)`

c) Назначение: возвращает вещественное число, извлечённое из строки `str`

Блок-схема:



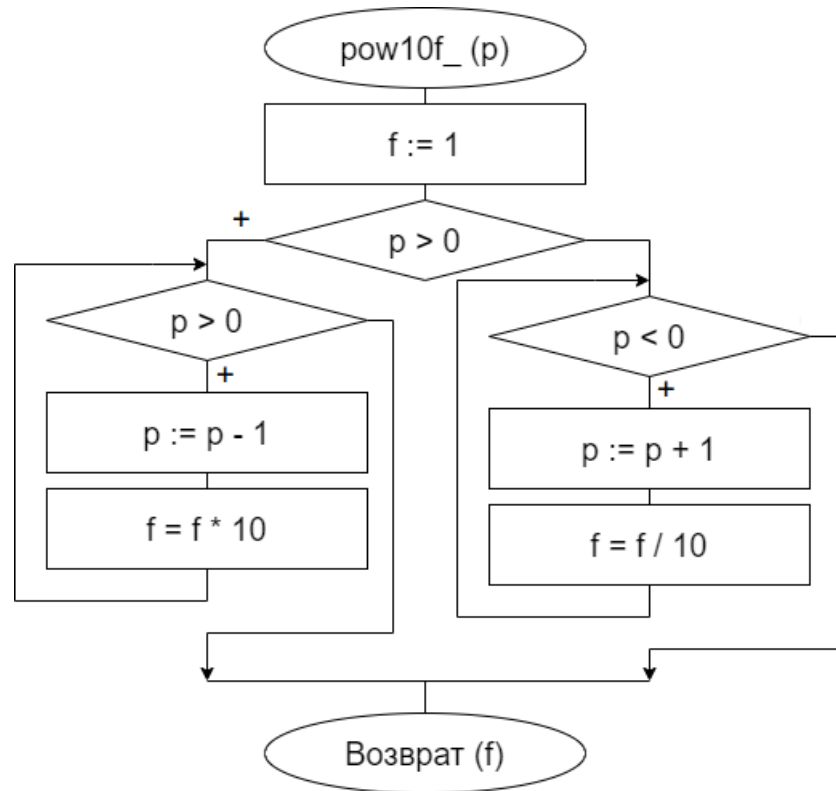


**Выделение подзадачи:** возвращение 10 в определённой степени

a) Заголовок: `pow10f_(p)`

b) Назначение: возвращает 10 в степени `p`

Блок-схема:



3) Запись целого числа в строку

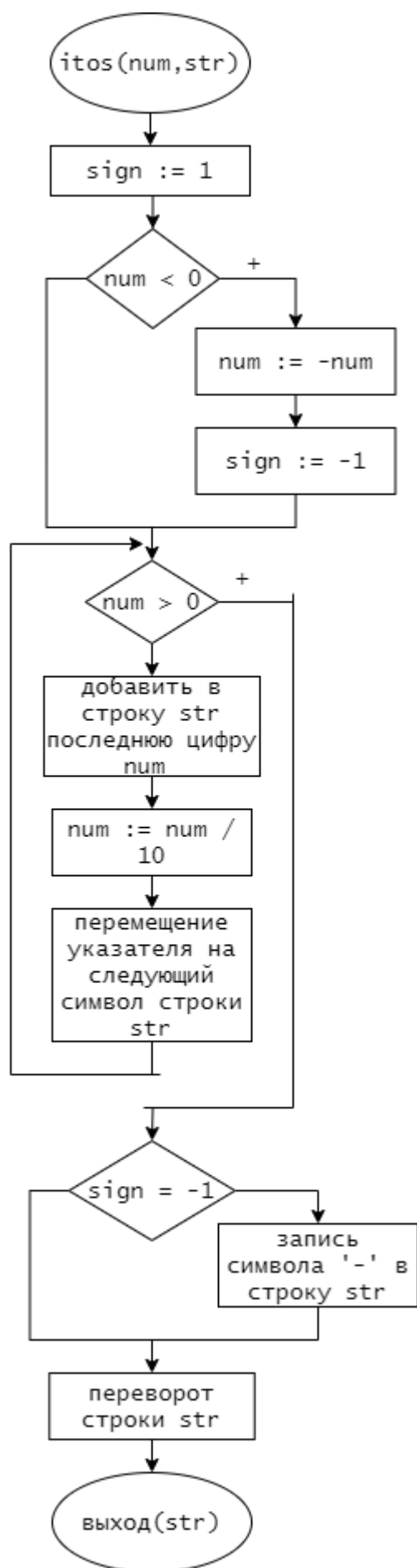
a) Выделение подзадачи

Переворот строки

b) Заголовок: `void itos(const int num, char str[])`

c) Назначение: записывает целое число `num` в строку `str`

Блок-схема:





**Выделение подзадачи:** переворот строки

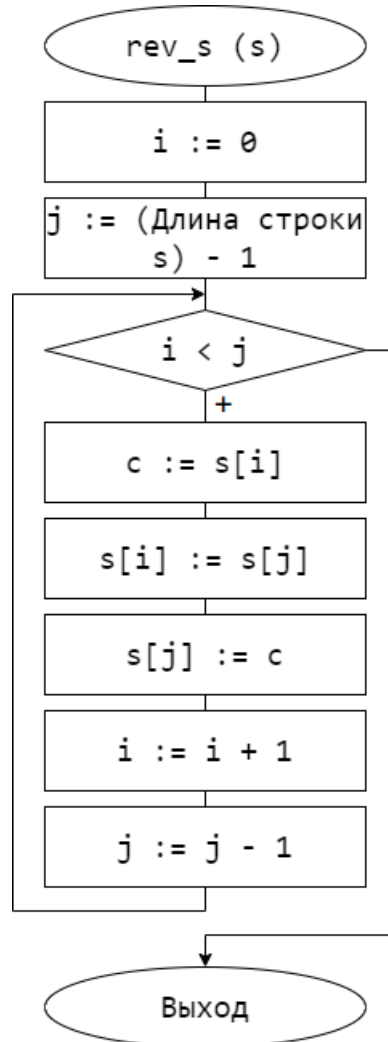
a) Выделение подзадачи:

Нахождение длины строки

b) Заголовок: `void rev_str(char *s)`

c) Назначение: переворачивает строку `str`

Блок-схема:



**Выделение подзадачи:** нахождение длины строки

a) Заголовок: `int slen_(char str[])`

b) Назначение: возвращает длину строки `str`

Блок-схема:



4) Перевод вещественного в строку

a) Выделение подзадач

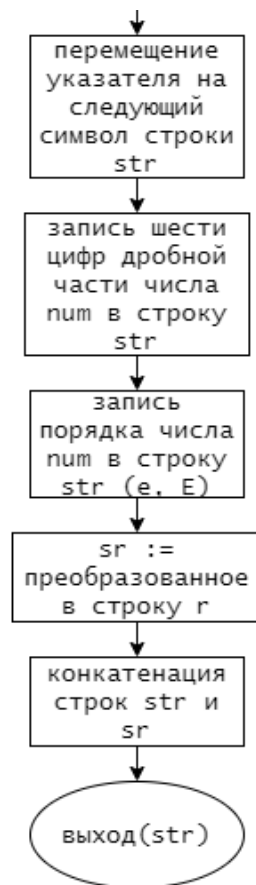
Получение нормальной формы числа, конкатенация строк

b) Заголовок: `void ftos(const double num, char str[])`

c) Назначение: записывает вещественное число `num` в строку `str`

Блок-схема:



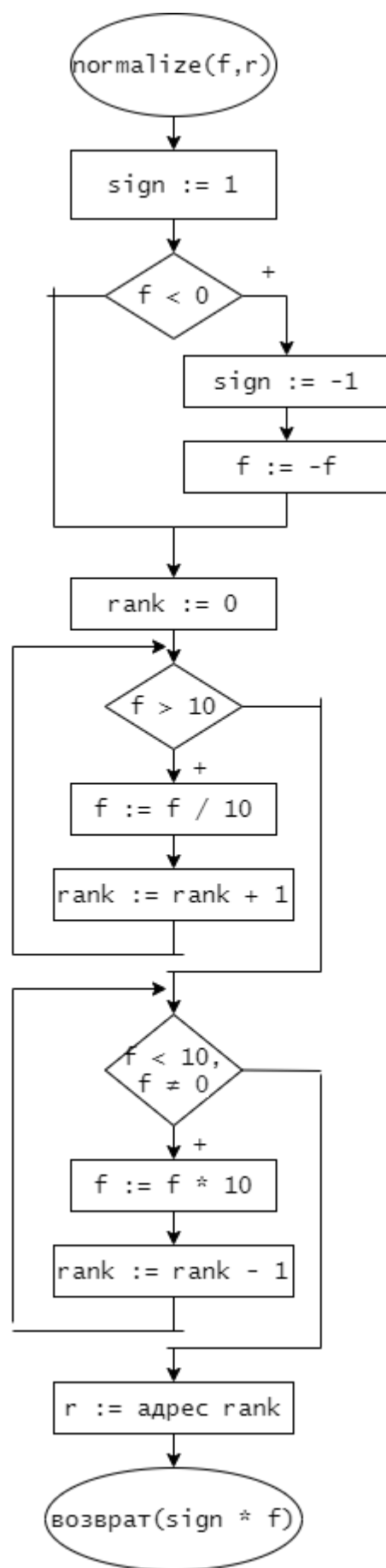


**Выделение подзадачи:** получение нормальной формы числа

a) Заголовок: `float normalize(float f, int* r)`

b) Назначение: возвращает  $f$  умноженное на  $10^k$ , такое что  $10 > f * 10^k \geq 1$  и записывает  $k$  в переменную по адресу  $r$

Блок-схема:

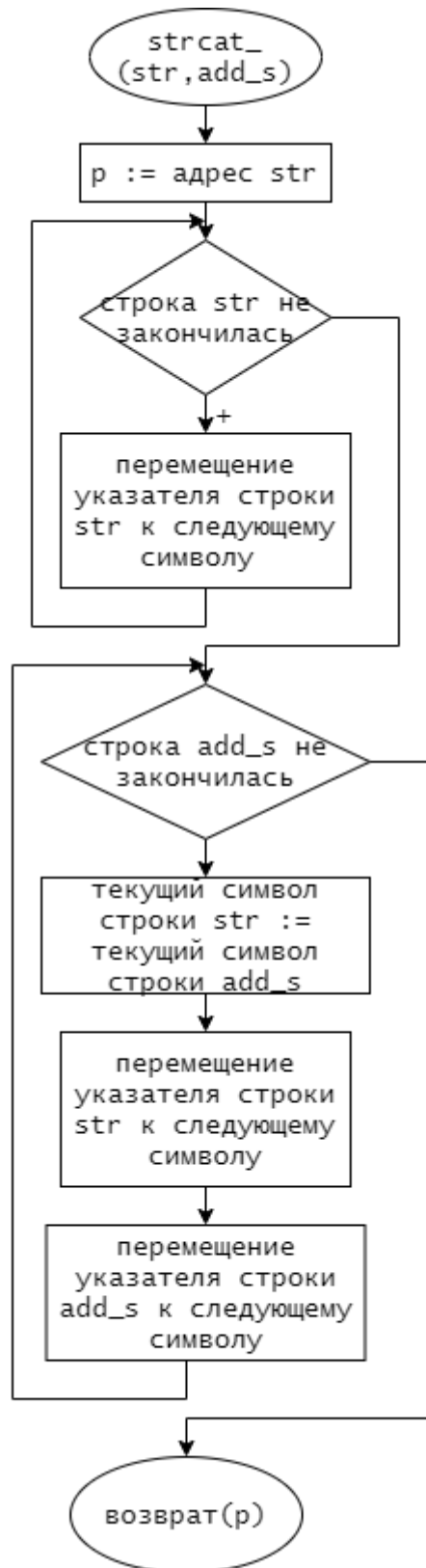


**Выделение подзадачи:** конкатенация строк

a) Заголовок: `char * strcat_(char *str, char *add_s)`

b) Назначение: возвращает указатель на начальный символ строки `s`, и приписывает в конец `s` строку `add_s`

Блок-схема:



#### 4. Тестовые данные

№ п/п	Выбранная функция	Вход	Выход
1	1	“123423”	123423
2	2	435432	“435432”
3	3	123.4325	“123.432500”
4	4	“5342.33”	5342.330000

#### 5. Текст программы

```
#include <stdio.h>
#include <windows.h>

/* возврат длины строки str */
int slen(char *str) {
    int i = 0;
    while (*str++ != '\0') i++;
    return (i);
}

/* переворот строки str */
void rev_str(char *str) {
    char c;
    int i = 0;
    int j = slen(str) - 1;
    while (i < j) {
        c = str[i];
        str[i] = str[j];
        str[j] = c;
        i++; j--;
    }
}

/* запись целого числа num в строку str */
char * itos(int num, char *str) {
    int sign = 1;
    char *p = str;

    if (num < 0) {
        num = -num;
        sign = -1;
    }

    while (num > 0) {
        *str++ = num % 10 + '0';
        num /= 10;
    }

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

```

    *str = '\\0';

    rev_str(p);
    return(p);
}

/* извлечение целого числа из строки str */
int stoi(const char* str) {
    while ((*str <= ' ') && (*str > '\\0'))
        ++str;

    int n = 0;
    int sign = 1;

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

    while ((*str >= '0') && (*str <= '9')) {
        n = n * 10 + *str - '0';
        ++str;
    }

    return (n * sign);
}

/* возведение 10 в степень p */
float powf_(int p) {
    float f = 1;

    if (p > 0)
        while (p != 0) {
            --p;
            f *= 10;
        }
    else
        while (p != 0) {
            ++p;
            f /= 10;
        }

    return f;
}

/* извлечение вещественного числа из строки str */
float stof(char* str) {
    while ((*str <= ' ') && (*str > '\\0'))
        ++str;

```



```

int sign = 1;

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

int a = 0;

while ((*str >= '0') && (*str <= '9')) {
    a = a * 10 + (*str - '0');
    ++str;
}

int k = 0;

if (*str == '.') {
    str++;
    while ((*str >= '0') && (*str <= '9')) {
        ++k;
        a = a * 10 + *str - '0';
        ++str;
    }
}

int p = 0;

if ((*str == 'e') || (*str == 'E')) {
    str++;

    while ((*str >= '0') && (*str <= '9')) {
        p = p * 10 + *str - '0';
        ++str;
    }
}

int l = p - k;

if (l < 0)
    return ((a * sign) / powf(-1));
else
    return (a * sign * powf(1));
}

/* конкатенация строк str и add_s */
char * strcat_(char *str, char *add_s) {
    char *p = str;

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

```

```

    while (*add_s != '\0') {
        *str = *add_s;
        ++str;
        ++add_s;
    }

    *str = '\0';
    return p;
}

/* получение нормальной формы вещественного числа f */
float normalize(float f, int* r) {
    float sign = 1;

    if (f < 0) {
        sign = -1;
        f = -f;
    }

    int rank = 0;

    while (f > 10) {
        f /= 10;
        ++rank;
    }

    while ((f < 10) && (f != 0)) {
        f *= 10;
        --rank;
    }

    *r = rank;
    return (sign * f);
}

/* запись вещественного числа f в строку str */
char * ftos(float f, char* str) {
    char *p = str;

    if (f < 0) {
        *str++ = '-';
        f = -f;
    }

    int r = 0;
    f = normalize(f, &r);

    int digit = (int) f;
    *str++ = digit + '0';
    f = f * 10;

```

```

        *str++ = '.';

        for (int i = 1; i < 6; ++i, f = f * 10) {
            int digit = (int)f % 10;
            *str++ = digit + '0';
        }

        *str++ = 'e';
        *str = '\\0';
        char sr[255];
        itos(r, sr);

        strcat_ (str, sr);
        return (p);
    }

int main() {
    system("chcp 65001");
    int num_i, v;
    float num_d;
    char s[255];

    printf("Выберите операцию:\\n"
           "1.Перевод целого в строку\\n"
           "2.Перевод вещественного в строку\\n"
           "3.Перевод строки в целое\\n"
           "4.Перевод строки в вещественное\\n");
    scanf("%d", &v);

    switch (v)
    {
        case 1:
            printf("\\nВведите целое число: ");
            scanf(" %i", &num_i);
            printf("Ваша строка с целым числом: \"%s\\\"",
itos(num_i, s));
            break;
        case 2:
            printf("\\nВведите вещественное число: ");
            scanf(" %f", &num_d);
            printf("Ваша строка с вещественным числом:
\\\"%s\\\"", ftos(num_d, s));
            break;
        case 3:
            printf("\\nВведите строку с целым числом: ");
            scanf("%s", s);
            printf("Ваше целое число: %i", stoi(s));
            break;
        case 4:
            printf("\\nВведите строку с вещественным числом:
");

```

```

scanf("%s", s);
printf("Ваше вещественное число: %f", stof(s));
break;
    }
}

```

## 6. Результаты работы

### *Пример №1*

```

1.Convert string to int
2.Convert int to string
3.Convert float to string
4.Convert string to float
1
Input string
123423
123423
Process finished with exit code 0

```

### *Пример №2*

```

1.Convert string to int
2.Convert int to string
3.Convert float to string
4.Convert string to float
2
Input int
435432
435432
Process finished with exit code 0

```

### *Пример №3*

```

1.Convert string to int
2.Convert int to string
3.Convert float to string
4.Convert string to float
3
Input float
123.4325
123.432500
Process finished with exit code 0

```

#### *Пример №4*

```
1.Convert string to int
2.Convert int to string
3.Convert float to string
4.Convert string to float
4
Input string
5342.33
5342.330000
Process finished with exit code 0
```

### **7. Анализ допущенных ошибок**

- нет