

Студент: Пантюхин А.Е., группа: ДТ-460а

Лабораторная работа №1

Задание:

Дано n различных символов. Напечатать все возможные перестановки этих символов.

Тестовые данные:

- 1) 123
- 2) 323
- 3) 4Eaf
- 4) 0

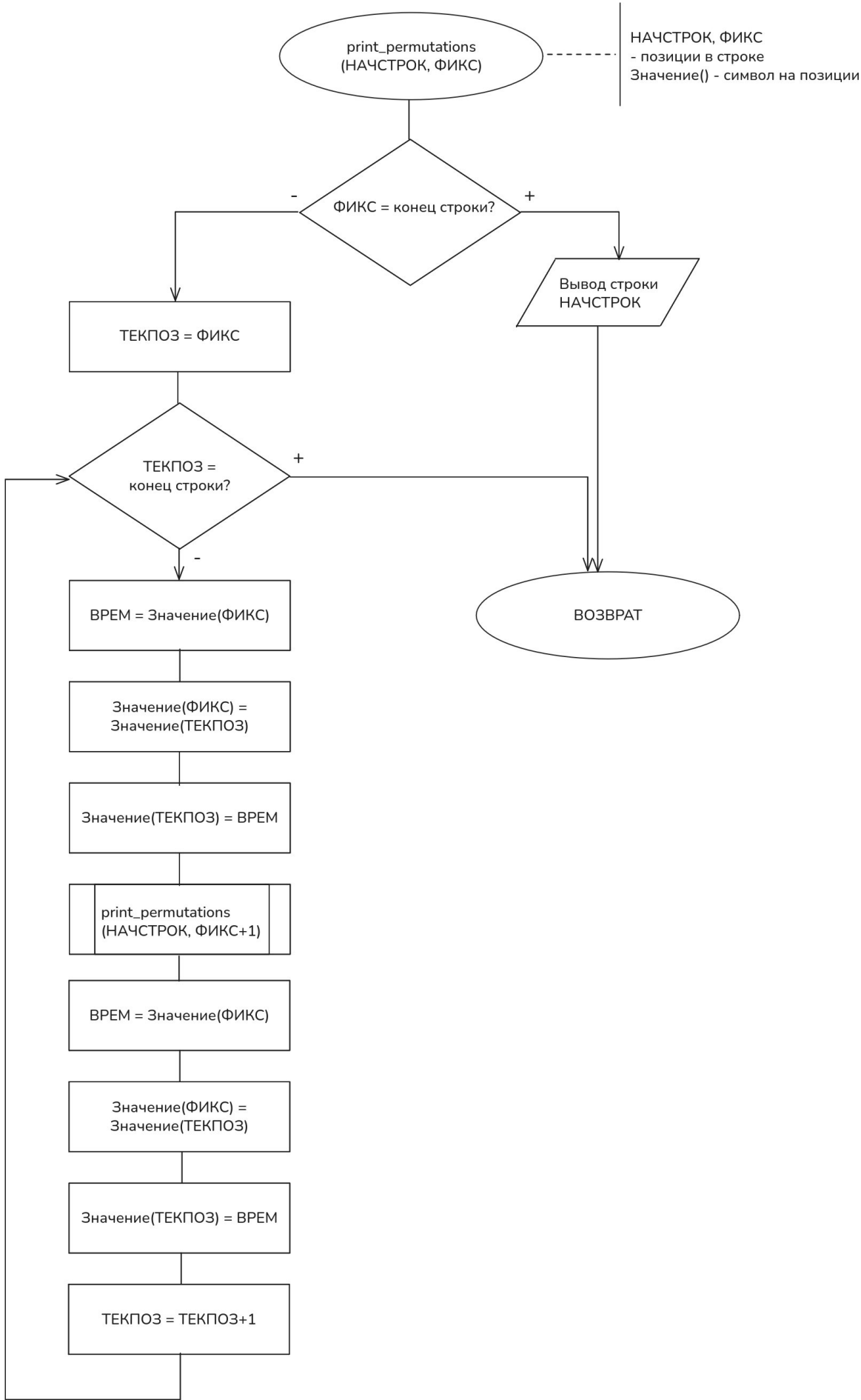
Словесное описание этапов решения задачи рекурсивным способом:

- 1) Функция принимает на вход два указателя, при первом вызове – одинаковых
- 2) Если входные данные не ведут на строки – выход
- 3) Первый указатель всегда обозначает начало строки, второй – текущий зафиксированный элемент
- 4) Если зафиксированный элемент – конец строки, то вывести строку
- 5) Иначе, начать движение по строке от зафиксированного указателя вправо
- 6) В цикле (пока не дошли до конца строки)
- 7) Меняем местами текущий элемент с текущим зафиксированным
- 8) Рекурсивный вызов функции, зафиксированный элемент – следующий от текущего зафиксированного
- 9) Обратно меняем местами значения зафиксированного и текущего элемента

В чём смысл:

- 1) Поочерёдно фиксируется каждый из элементов строки (рекурсивные вызовы)
- 2) Поочерёдно меняется каждый из элементов с зафиксированным (внутри цикла)
- 3) Таким образом, в рамках вызовов, сначала выведется неизменённая строка
- 4) Потом строка с перестановкой последних двух элементов
- 5) Далее третий с конца элемент поменяется местами со вторым с конца
- 6) Аналогично, строка с перестановкой последних двух
- 7) И так далее.

Блок-схема алгоритма рекурсивной функции:



Текст программы с комментариями:

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

//вспомогательная функция смены значений местами.
void swap_values(char* ptr1, char* ptr2)
{
    char buff = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = buff;
    return;
}

//вывод перестановок
//source - указатель на начало строки, всегда
//locked_ptr - указатель на "зафиксированный" элемент
int print_permutations(char* const source, char* const locked_ptr)
{
    char* move_ptr = locked_ptr; //указатель на адрес, начиная от текущего
    зафиксированного
    //если на входе что-то не так - вернём ошибку
    if (!source || !locked_ptr) return 0;

    //если служебный указатель дошёл до конца строки - выведем итоговую строку
    if (!*locked_ptr) printf("%s\n", source);
    //начиная с позиции служебного указателя(при первом вызове - 0, далее - по
    элементам строки)
    //пока строка не кончилась
    while (*move_ptr)
    {
        //поменяем местами значения по указателям (при первом вызове ничего не
        изменится)
        swap_values(locked_ptr, move_ptr);
        //рекурсивный вызов со сдвигом постоянной позиции на 1 (позиции
        "зафиксированного" элемента)
        print_permutations(source, locked_ptr+1);
        //поменяем местами обратно для следующего вызова.
        swap_values(locked_ptr, move_ptr);
        //двигаемся по строке далее
        move_ptr++;
    }
    return 1;
}

int main() {
    setlocale(LC_ALL, "Russian");
    // BUFSIZ - директива препроцессора из библиотеки stdio, определяющая размер
    буфера потока.
```

```
char user_input[BUFSIZ] = {0};  
printf("Введите символы, не более %d:\n", BUFSIZ);  
scanf("%s", user_input);  
//или fgets(user_input, BUFSIZ-1, stdin);  
print_permutations(user_input, user_input);  
return 0;  
}
```