

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 №10

по дисциплине: Основы программирования
тема: «Рекурсивные функции»

Выполнил: студент группы ПВ-223
Мелехов Артём Дмитриевич

Проверили:
ст. преп. Притчин Иван Сергеевич
асс. Черников Сергей Викторович

Белгород 2022 г.

Лабораторная работа №10 «Рекурсивные функции»

Цель работы: получение навыков написания рекурсивных функций.

Содержание отчета:

Тема лабораторной работы

Цель лабораторной работы

Решения задач варианта 6. Для каждой задачи указаны:

- Условие задачи.
- Тестовые данные.
- Исходный код с комментариями (спецификацией) к функциям.

Вывод.

Задача №3.

Условие:

Дан знаменатель и первый член геометрической прогрессии. Вычислить n -й член прогрессии.

Тестовые данные:

Входные данные	Выходные данные
1 2 4	8
4 2 4	32
1 3 1	1
4 2 3	16

Код функции:

```
//Вычисление n-ного члена геометрической прогрессии
long long the_n_term_of_the_geometric_progression
    (const long long first_term,
     const long long denominator, long long n)
{
    if ((n--) == 1)
        return first_term;

    return the_n_term_of_the_geometric_progression
        (first_term, denominator, n) * denominator;
}
```

Задача №4.

Условие:

Дана упорядоченная по убыванию последовательность целых чисел, записанная в массив. Определить, есть ли среди членов данной последовательности число x , и если есть, найти номер этого члена. Бинарным поиском не пользоваться.

Тестовые данные:

Входные данные	Выходные данные
{10, 9, 8, 2, 0}, 3	-1
{10, 9, 8, 2, 0}, 2	3
{10, 5, 4, 4, 7, 8}, 10	0

Код программы:

```
long long search_in_sorted_sequence_
(const long long * const a, const long long x,
 const long long i)
{
    if (a[i] <= x)
        return i;

    return search_in_sorted_sequence_(a, x, i + 1);
}

//Поиск в упорядоченном массиве элемента x
long long search_in_sorted_sequence(long long* a,
 const long long x)
{
    long long index = search_in_sorted_sequence_(a, x, 0);

    return a[index] == x ? index : -1;
}
```

Задача №7.

Условие:

Даны натуральные числа a и b . Определить, могут ли эти числа быть соседними членами последовательности Фибоначчи. Последовательность Фибоначчи задаётся следующим образом:

$$f_1 = f_2 = 1 \quad f_i = f_{i-1} + f_{i-2} \text{ для } i > 2$$

Тестовые данные:

Входные данные	Выходные данные
1 1	true
0 1	false
7 9	false
13 21	true

Код программы:

```
#include <stdbool.h>

bool is_fib_sum(long long sum_ab, long long acc_sum, long long last_fib2, long long last_fib1)
{
    acc_sum += last_fib1;

    if (acc_sum >= sum_ab)
        return acc_sum == sum_ab;

    return is_fib_sum(sum_ab, acc_sum, last_fib1, last_fib1 + last_fib2);
}

//Возвращает 'истину', если 2 числа являются соседними
//числами последовательности Фибоначчи
bool is_fib_sum(long long a, long long b)
{
    return is_fib_sum(a + b, 1, 0, 1);
}
```

Задача №12.

Условие:

**** Реализовать сортировку выбором.**

Тестовые данные:

Входные данные	Выходные данные
16 8 4 2 1	1 2 4 8 16
2 2 2 2 2	2 2 2 2 2
0	0
1 2 3 4	1 2 3 4

Код программы:

```
long long min_index(long long* a, long long i,
    long long j)
{
    if (i == j)
        return;

    long long minimal_index = min_index(a, i + 1, j);

    return a[i] < a[minimal_index] ? i : minimal_index;
}

void swap(long long* a, long long* b)
{
    long long t = *a;
    *a = *b;
    *b = t;
}

void recursive_sorting_by_choice_(long long* a,
    size_t size, long long index)
{
    if (index == size)
        return;

    long long minimal_index = min_index(a, index,
        size - 1);

    if (minimal_index != index)
        swap(&a[minimal_index], &a[index]);

    recursive_sorting_by_choice_(a, size, index + 1);
}

//Рекурсивная сортировка выбором
void recursive_sorting_by_choice(long long* a,
    size_t size)
{
    recursive_sorting_by_choice_(a, size, 0);
}
```

Вывод: в ходе работы были получены навыки написания рекурсивных функций.