

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 №11

по дисциплине: Основы программирования
тема: «Рекурсивные функции»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
Притчин Иван Сергеевич
Черников Сергей Викторович

Код-ревьюер: ст. группы ПВ-223
Голуцкий Георгий Юрьевич

Белгород 2022 г.

Лабораторная работа № 11

Вариант №1

Содержание отчёта:

- Тема лабораторной работы.
- Цель лабораторной работы.
- Решения задач.
 - Условие задачи.
 - Исходный код рекурсивных функций без спецификаций.
- Вывод по работе.

Тема лабораторной работы: Рекурсивные функции

Цель лабораторной работы: получение навыков написания рекурсивных функций.

Решения задач:

1. Вывести данное натуральное число в восьмеричной системе счисления.

2.c

```
#include "../libs/alg/alg.h"

int main() {
    unsigned long long num;
    scanf("%llu", &num);

    printAsOct(num);
}
```

2func.c

```
#include "../alg.h"

void printAsOct_(unsigned long long num) {
    if (num == 0)
        return;
    else {
        int digit = num % 8;
        printAsOct(num / 8);
        printf("%d", digit);
    }
}

void printAsOct(unsigned long long num) {
    if (num == 0)
        printf("0");
    else
        printAsOct_(num);
}
```

2. 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.

4.c

```
#include <stdlib.h>

#include "../libs/alg/alg.h"

int main() {
    int arraySize;
    scanf("%d", &arraySize);

    int *a = (int*) malloc(sizeof(int) * arraySize);

    inputArray(a, arraySize);

    int x;
    scanf("%d", &x);

    printf("%d", linearRecSearchInArray(a, arraySize, x));

    free(a);

    return 0;
}
```

4func.c

```
#include "../alg.h"

size_t linearRecSearch_(const int * const array, size_t currentIndex,
                        size_t arraySize, int searchElement) {
    if (currentIndex == arraySize || array[currentIndex] < searchElement)
        return -1;
    else
        return array[currentIndex] == searchElement ?
            currentIndex :
            linearRecSearch_(array, currentIndex + 1, arraySize, searchElement);
}

size_t linearRecSearchInArray(const int * const array, size_t arraySize,
                              int searchElement) {
    return linearRecSearch_(array, 0, arraySize, searchElement);
}
```

3. С клавиатуры вводятся положительные вещественные числа a_1, a_2, \dots, a_n . Признаком конца ввода – отрицательное число. Вывести следующие значения:

$$\frac{a_n + a_{n-1}}{2}, \frac{a_{n-1} + a_{n-2}}{2}, \dots, \frac{a_2 + a_1}{2}$$

10.c

```
#include "../libs/alg/alg.h"

int main() {
    size_t elementsAmount;
    scanf("%zu", &elementsAmount);

    printElementsOfSequence(elementsAmount);

    return 0;
}
```

10func.c

```
#include "../alg.h"

void printElementsOfSequence_(int lastElement, size_t size) {
    if (size == 0)
        return;
    else {
        int currentElement;
        scanf("%d", &currentElement);

        printElementsOfSequence_(currentElement, size - 1);

        double element = (lastElement + currentElement) / 2.0;

        printf("%.2lf ", element);
    }
}

void printElementsOfSequence(size_t size) {
    if (size != 0) {
        int curElement;
        scanf("%d", &curElement);

        printElementsOfSequence_(curElement, size - 1);
    }
}
```

4. ** Реализовать сортировку выбором

12.c

```
#include <stdlib.h>

#include "../libs/alg/alg.h"

int main() {
    ssize_t arraySize;
    scanf("%zd", &arraySize);

    int *array = (int *) malloc(sizeof(int) * arraySize);
    inputArray(array, arraySize);

    selectionSort(array, arraySize);

    outputArray(array, arraySize);

    free(array);

    return 0;
}
```

12func.c

```
#include "../alg.h"

ssize_t getFirstMinElementIndex_(const int * const array, ssize_t currentIndex,
                                ssize_t arraySize) {
    if (currentIndex == arraySize - 1)
        return currentIndex;
    else {
        ssize_t nextMinIndex = getFirstMinElementIndex_(array, currentIndex + 1,
                                                         arraySize);
        if (array[nextMinIndex] < array[currentIndex])
            return nextMinIndex;
        else
            return currentIndex;
    }
}

ssize_t getFirstMinElementIndex(const int * const array, ssize_t arraySize) {
    return getFirstMinElementIndex_(array, 0, arraySize);
}

void selectionSort(int * const array, ssize_t arraySize) {
    if (arraySize <= 1)
        return;
    else {
        ssize_t minIndex = getFirstMinElementIndex(array, arraySize);
        intSwap(array, array + minIndex);
        selectionSort(array + 1, arraySize - 1);
    }
}
```

Вывод: в ходе выполнения лабораторной работы получены навыки написания рекурсивных функций.