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-ревьер: 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);  
}

1. **С клавиатуры вводятся положительные вещественные числа 𝑎1, 𝑎2, . . . , 𝑎𝑛. Признак конца ввода – отрицательное число. Вывести следующие значения:**

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);  
 }  
}

1. **\*\* 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.