

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 №8**

по дисциплине: Основы программирования

тема: «Реализация функций

для работы с одномерными массивами в стиле С»

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 № 8

## Вариант №1

### Содержание отчёта:

- Тема лабораторной работы.
- Номер варианта.
- Цель лабораторной работы.
- Решения задач.
- Вывод по работе.

**Тема лабораторной работы:** Реализация функций для работы с одномерными массивами в стиле C

**Цель лабораторной работы:** получение навыков написания функций при решении задач на одномерные массивы.

### Решения задач:

#### 1. Поиск позиции элемента с начала массива (по функции-предикату).

```
#include <stdbool.h>
#include <corecrt.h>

/* возвращает позицию первого с начала элемента массива array размера arraySize
 * удовлетворяющему условию выраженному функцией-предикатом predicate
 */
size_t findByPredicateFromBegin(const int *array, const size_t arraySize,
                               bool (*predicate)(int)) {
    size_t index = 0;
    while (index < arraySize && !predicate(array[index]))
        index++;

    return index;
}
```

#### 2. Поиск позиции с конца массива (по функции-предикату).

```
#include <stdbool.h>

/* возвращает позицию первого с конца элемента массива array размера arraySize
 * удовлетворяющему условию выраженному функцией-предикатом predicate
 */
int findByPredicateFromEnd(const int *array, const int arraySize,
                           bool (*predicate)(int)) {
    int index = arraySize - 1;
    while (index >= 0 && !predicate(array[index]))
        index--;

    return index;
}
```

### 3. Вычисление количества элементов массива, удовлетворяющих функции-предикату.

```
#include <corecrt.h>

/* возвращает количество элементов массива array размера arraySize
 * удовлетворяющих условию выраженному функцией-предикатом predicate
 */
int countByPredicate(const int *array, const size_t arraySize,
                    bool (*predicate)(int)) {
    int count = 0;
    for (size_t i = 0; i < arraySize; i++)
        if (predicate(array[i]))
            count++;

    return count;
}
```

### 4. Реализуйте циклический сдвиг массива влево на $k$ позиций.

```
#include <corecrt.h>

// выполняет циклический сдвиг влево массива array размера arraySize на k
void leftShift(int *array, const size_t arraySize, const size_t k) {
    int tempArray[arraySize];
    for(int i = 0; i < arraySize; i++)
        tempArray[i] = array[(i + k) % arraySize];

    moveArray(array, tempArray, arraySize);
}

/* копирует данные из массива data размером arraySize в
 * массив dest размером arraySize
 */
void moveArray(int *dest, const int *data, const size_t arraySize) {
    for(int i = 0; i < arraySize; i++)
        dest[i] = data[i];
}
```

### 5. Реализуйте функцию *forEach*, которая применяет функцию $f$ к элементам массива $a$ размера $size$ .

```
#include <corecrt.h>

// применяет к каждому элементу массива a размером size функцию f
void forEach(int *a, const size_t size, void (*f)(int *)) {
    for(int i = 0; i < size; i++)
        f(a + i);
}
```

6. Реализуйте функцию *any*, которая возвращает значение 'истина', если хотя бы один элемент массива *a* размера *size* удовлетворяют функции-предикату *f*, иначе – 'ложь'.

```
#include <corecrt.h>

/* возвращает "истину" если хотя бы один элемент массива a размера size
 * удовлетворяет условию выраженному функцией предикатом f, иначе - "ложь"
 */
bool any(const int *a, const size_t size, bool (*f)(int)) {
    for (size_t i = 0; i < size; i++)
        if (f(a[i]))
            return true;

    return false;
}
```

7. Реализуйте функцию *all*, которая возвращает значение 'истина', если все элементы массива *a* размера *size* удовлетворяют функции-предикату *f*, иначе – 'ложь'.

```
#include <corecrt.h>

/* возвращает "истину" если все элементы массива a размера size
 * удовлетворяют условию выраженному функцией предикатом f, иначе - "ложь"
 */
bool all(const int *a, const size_t size, bool (*f)(int)) {
    for (size_t i = 0; i < size; i++)
        if (!f(a[i]))
            return false;

    return true;
}
```

8. Реализуйте функцию *arraySplit*, которая разделяет элементы массива *a* размера *size* на элементы, удовлетворяющие функции-предикату *f*, сохраняя в массиве *b*, иначе – в массиве *c*.

```
#include <corecrt.h>

/* распределяет элементы массива a размера size таким образом, что элементы,
 * удовлетворяющие условию выраженному функцией-предикатом f, добавляются в
 * массив b, иначе - в массив c
 */
void arraySplit(const int *a, const size_t size, bool (*f)(int), int *b, int *c) {
    for (size_t i = 0, bInd = 0, cInd = 0; i < size; i++)
        if (f(a[i]))
            b[bInd++] = a[i];
        else
            c[cInd++] = a[i];
}
```

**Вывод:** в ходе выполнения лабораторной работы получены навыки написания функций при решении задач на одномерные массивы.