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-ревьер: 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 С

**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;  
}

1. **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;  
}

1. 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;  
}

1. 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];  
}

1. **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);  
}

1. **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**;  
}

1. **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**;  
}

1. **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.