

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: «Решение задач на одномерные массивы с использованием функций»

Выполнил: студент группы ПВ-223
Мелехов Артём Дмитриевич

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

Белгород 2022 г.

Лабораторная работа № «Решение задач на одномерные массивы с использованием функций»

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

Содержание отчета:

Тема лабораторной работы

Цель лабораторной работы

Решения задач. Для каждой задачи указаны:

- Условие задачи.
- Исходный код функции и её спецификация в виде строки-комментария.

Вывод.

Задача №1.

Условие:

Ввод массива а размера n.

Код функции:

```
//Ввод массива а размера n
void inputArray(long long* a, const size_t n)
{
    for (size_t i = 0; i < n; i++)
        scanf("%lld", &a[i]);
}
```

Задача №2.

Условие:

Вывод массива а размера n.

Код функции:

```
//Вывод массива а размера n
void outputArray(const int* a, const size_t n)
{
    for (size_t i = 0; i < n; i++)
        printf("%d ", a[i]);

    printf("\n");
}
```

Задача №3.

Условие:

Поиск позиции элемента со значением x с начала массива:

Входные данные	Выходные данные
$a = \{6, 4, 5\}$ $x = 4$	1
$a = \{1, 2, 3, 4\}$ $x = 1$	0
$a = \{1, 2, 4, 4\}$ $x = 4$	2
$a = \{4, 2, 4\}$ $x = 0$	-1

Код функции:

```
//Поиск позиции элемента со значением  $x$  с начала массива
size_t getFirstXIndex(int *a, const size_t n, int x)
{
    for (size_t i = 0; i < n; i++)
        if (a[i] == x)
            return i;

    return -1;
}
```

Задача №4.

Условие:

Поиск позиции первого отрицательного элемента.

Код функции:

```
//Поиск позиции первого отрицательного элемента
size_t getFirstNegativeIndex(int *a, const size_t n)
{
    for (size_t i = 0; i < n; i++)
        if (a[i] < 0)
            return i;

    return -1;
}
```

Задача №6.

Условие:

Поиск позиции последнего чётного элемента.

Код функции:

```
//Поиск позиции последнего четного элемента
int getLastEvenIndex(int *a, const size_t n)
{
    for (int i = n - 1; i >= 0; i--)
        if (a[i] % 2 == 0)
            return i;

    return -1;
}
```

Задача №8.

Условие:

Вычисление количества отрицательных элементов.

Код функции:

```
//Подсчёт количества отрицательных элементов
long long countNegative(int *a, const size_t n)
{
    long long negativeCount = 0;

    for (size_t i = 0; i < n; i++)
        negativeCount += a[i] < 0;

    return negativeCount;
}
```

Задача №10.

Условие:

Изменение порядка элементов массива на обратный.

Код функции:

```
//Изменение порядка элементов массива на обратный
void reverseArray(int *a, const size_t n)
{
    for (size_t i = 0, j = n - 1; i < j; i++, j--)
        swap(&a[i], &a[j]);
}
```

Задача №11.

Условие:

Проверка, является ли последовательность палиндромом.

Код функции:

```
#include <stdbool.h>

//Проверка массива на палиндром
bool isArrayPoly(int *a, const size_t n)
{
    for (size_t i = 0, j = n - 1; i < j; i++, j--)
        if (a[i] != a[j])
            return false;

    return true;
}
```

Задача №12.

Условие:

Сортировка массива выбором.

Код функции:

```
//Сортировка массива выбором
void sortingByChoice(int *a, const size_t n)
{
    for (int i = 0; i < n; i++)
    {
        int max = a[i];
        int index = i;

        for (int j = i + 1; j < n; j++)
            if (a[j] > max)
            {
                max = a[j];
                index = j;
            }

        if (i != index)
            swap(&a[i], &a[index]);
    }
}
```

Задача №13.

Условие:

Удаление из массива всех нечётных элементов.

Код функции:

```
//Удаление всех нечётных элементов массива
void sortingByChoice(int *a, size_t *n)
{
    size_t iRead = 0;

    while (iRead < *n && a[iRead] % 2 == 0)
        iRead++;

    size_t iWrite = iRead;

    while (iRead < *n)
    {
        if (a[iRead] % 2 == 0)
        {
            a[iWrite] = a[iRead];
            iWrite++;
        }

        iRead++;
    }

    *n = iWrite;
}
```

Задача №14.

Условие:

Вставка элемента в массив с сохранением относительного порядка других элементов.

Код функции:

```
//Вставка элемента в массив с сохранением относительного порядка других элементов
void insertingAnElementIntoAnArray(int *a, size_t *n, const size_t position, const
int value)
{
    for (size_t i = *n - 1; i >= position; i--)
        a[i + 1] = a[i];

    a[position] = value;
    (*n)++;
}
```

Задача №15.

Условие:

Добавление элемента в конец массива.

Код функции:

```
//Вставка элемента в конец массива
void addingAnElementToTheEndOfTheArray(int *a, size_t *n, int value)
{
    a[*n] = value;
    (*n)++;
}
```

Задача №16.

Условие:

Удаление элемента с сохранением относительного порядка других элементов.

Код функции:

```
//Удаление элемента с сохранением относительного порядка других элементов
void deletinAnItemWhileMaintainingOrder(int *a, size_t *n, size_t position)
{
    for (size_t i = position; i < *n; i++)
        a[i] = a[i + 1];

    (*n)--;
}
```

Задача №17.

Условие:

Удаление элемента без сохранением относительного порядка других элементов.

Код функции:

```
//Удаление элемента без сохранения относительного порядка других элементов
void deletingAnItemWithoutPreservingTheOrder(int* a, size_t* n, size_t position)
{
    (*n)--;
    a[position] = a[*n];
}
```

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