

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 №0
по дисциплине: «Дискретная математика»

Выполнил/а: ст. группы ПВ-231
Чупахина София Александровна

Проверили:
Островский Алексей Мичеславович
Рязанов Юрий Дмитриевич

Белгород, 2023

Библиотека «array_input_output»:	3
Задача 1:	3
Задача 2:	4
Задача 3:	5
Задача 4:	6
Задача 5:	7
Задача 6:	8
Задача 7:	9
Задача 8:	9
Задача 9:	10
Задача 10:	11
Задача 11:	12
Задача 12:	13
Функция main с вводом данных и выводом результатов работы функций с решением задачи:	14

Библиотека «array_input_output»:

```
#include <stdio.h>
```

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

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

Задача 1:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Получить массив C, содержащий все элементы массивов A и B без повторений.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	C = {2, 3, 1, 5, 4, 6}	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {1, 2}	C = {2, 3, 1, 5, 4}	Неупорядоченные массивы, один из которых не имеет уникальных элементов
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	C = {2, 3, 1, 5, 4, 6, 9, 7}	Неупорядоченные массивы, у которых общих элементов нет

Выполнение задания:

```
void getJoinArray(const int *a, const int a_len, const int *b,
const int b_len, int *c, int *c_len) {
    for (int a_index = 0; a_index < a_len; a_index++) {
        c[a_index] = a[a_index];
    }
    *c_len = a_len;

    for (int b_index = 0; b_index < b_len; b_index++) {
```

```

        int is_element_unique = 1;
        for (int a_index = 0; a_index < a_len; a_index++) {
            if (b[b_index] == a[a_index]) {
                is_element_unique = 0;
                break;
            }
        }

        if (is_element_unique) {
            c[*c_len] = b[b_index];
            *c_len += 1;
        }
    }
}

```

Задача 2:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Получить массив C, содержащий все такие элементы, которые есть и в массиве A и в массиве B.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	C = {2, 1}	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {1, 2}	C = {2, 1}	Неупорядоченные массивы, один из которых не имеет уникальных элементов
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	C = {}	Неупорядоченные массивы, у которых общих элементов нет

Выполнение задания:

```

void getIntersectionArray(const int *a, const int a_len, const int
*b, const int b_len, int *c, int *c_len) {
    *c_len = 0;
    for (int a_index = 0; a_index < a_len; a_index++) {
        for (int b_index = 0; b_index < b_len; b_index++) {
            if (a[a_index] == b[b_index]) {
                c[*c_len] = a[a_index];
                *c_len += 1;
                break;
            }
        }
    }
}

```

```

    }
  }
}

```

Задача 3:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Получить массив C, содержащий все элементы массива A, которых нет в B.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	C = {3, 5, 4}	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	C = {2, 3, 1, 5, 4}	Неупорядоченные массивы, у которых общих элементов нет
A = {2, 3, 1, 5, 4}; B = {1, 2}	C = {3, 5, 4}	Неупорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {2, 3, 1, 5, 4}	C = {}	Неупорядоченные массивы, причем массив A является подмассивом B

Выполнение задания:

```

void getExceptionArrayByA(int *a, int a_len, int *b, int b_len,
int *c, int *c_len) {
    *c_len = 0;
    for (int a_index = 0; a_index < a_len; a_index++) {
        int is_element_unique = 1;
        for (int b_index = 0; b_index < b_len; b_index++) {
            if (b[b_index] == a[a_index]) {
                is_element_unique = 0;
                break;
            }
        }
        if (is_element_unique) {
            c[*c_len] = a[a_index];
            *c_len += 1;
        }
    }
}

```

Задача 4:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Получить массив C, содержащий все элементы массива A, которых нет в B и все элементы массива B, которых нет в A.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	C = {3, 5, 4, 6}	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	C = {2, 3, 1, 5, 4, 6, 9, 7}	Неупорядоченные массивы, у которых общих элементов нет
A = {2, 3, 1, 5, 4}; B = {1, 2}	C = {3, 5, 4}	Неупорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {2, 3, 1, 5, 4}	C = {3, 5, 4}	Неупорядоченные массивы, причем массив A является подмассивом B
A = {1, 2}; B = {2, 1};	C = {}	Повторяющие друг друга массивы

Выполнение задания:

```
void getExceptionArray(const int *a, const int a_len, const int *b,
const int b_len, int *c, int *c_len) {
    *c_len = 0;
    for (int a_index = 0; a_index < a_len; a_index++) {
        int is_element_unique = 1;
        for (int b_index = 0; b_index < b_len; b_index++) {
            if (b[b_index] == a[a_index]) {
                is_element_unique = 0;
                break;
            }
        }
        if (is_element_unique) {
            c[*c_len] = a[a_index];
            *c_len += 1;
        }
    }
}
```

```
    for (int b_index = 0; b_index < b_len; b_index++) {
        int is_element_unique = 1;
        for (int a_index = 0; a_index < a_len; a_index++) {
            if (b[b_index] == a[a_index]) {
                is_element_unique = 0;
                break;
            }
        }
    }
```

```

    }
    if (is_element_unique) {
        c[*c_len] = b[b_index];
        *c_len += 1;
    }
}

```

Задача 5:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Определить, верно ли, что массив B содержит каждый элемент массива A.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	0	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	0	Неупорядоченные массивы, у которых общих элементов нет
A = {2, 3, 1, 5, 4}; B = {1, 2}	0	Неупорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {2, 3, 1, 5, 4}	1	Неупорядоченные массивы, причем массив A является подмассивом B
A = {1, 2}; B = {2, 1};	1	Повторяющие друг друга массивы

Выполнение задания:

```

int isAPartOfB(const int *a, const int a_len, const int *b, const
int b_len) {
    int is_a_part_of_b = 1;
    for (int a_index = 0; a_index < a_len; a_index++) {
        int is_element_in_b = 0;
        for (int b_index = 0; b_index < b_len; b_index++) {
            if (b[b_index] == a[a_index]) {
                is_element_in_b = 1;
                break;
            }
        }
        if (!is_element_in_b) {
            is_a_part_of_b = 0;
            break;
        }
    }
}

```

```

    return is_a_part_of_b;
}

```

Задача 6:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Определить, верно ли, что массивы A и B состоят из одинаковых элементов.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	0	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	0	Неупорядоченные массивы, у которых общих элементов нет
A = {2, 3, 1, 5, 4}; B = {1, 2}	0	Неупорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {2, 3, 1, 5, 4}	0	Неупорядоченные массивы, причем массив A является подмассивом B
A = {1, 2}; B = {2, 1};	1	Повторяющие друг друга массивы

Выполнение задания:

```

int isArrayEqual(const int *a, const int a_len, const int *b,
const int b_len) {
    int is_a_part_of_b = 1;
    for (int a_index = 0; a_index < a_len; a_index++) {
        int is_element_in_b = 0;
        for (int b_index = 0; b_index < b_len; b_index++) {
            if (b[b_index] == a[a_index]) {
                is_element_in_b = 1;
                break;
            }
        }
        if (!is_element_in_b) {
            is_a_part_of_b = 0;
            break;
        }
    }
    return is_a_part_of_b && a_len == b_len;
}

```

Задача 7:

Текст задания:

Дано:

A – массив натуральных чисел, в котором нет одинаковых элементов;

B – массив натуральных чисел, в котором нет одинаковых элементов.

Определить, верно ли, что в массивах A и B нет общих элементов.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {2, 3, 1, 5, 4}; B = {1, 6, 2}	0	Неупорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {2, 3, 1, 5, 4}; B = {6, 9, 7}	1	Неупорядоченные массивы, у которых общих элементов нет
A = {2, 3, 1, 5, 4}; B = {1, 2}	0	Неупорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {2, 3, 1, 5, 4}	0	Неупорядоченные массивы, причем массив A является подмассивом B
A = {1, 2}; B = {2, 1};	0	Повторяющие друг друга массивы

Выполнение задания:

```
int isArraysDisjoint(const int *a, const int a_len, const int *b,
const int b_len) {
    int is_arrays_disjoint = 1;
    for (int a_ind = 0; a_ind < a_len; a_ind++) {
        for (int b_ind = 0; b_ind < b_len; b_ind++) {
            if (b[b_ind] == a[a_ind]) {
                is_arrays_disjoint = 0;
                break;
            }
        }
    }
    return is_arrays_disjoint;
}
```

Задача 8:**Текст задания:**

Даны массивы натуральных чисел A и B, упорядоченные по возрастанию.

Получить упорядоченный по возрастанию массив C, содержащий все элементы массивов A и B.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {1, 2, 3, 4, 5}; B = {1, 2, 6}	C = {1, 2, 3, 4, 5, 6}	Упорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы

A = {1, 2, 3, 4, 5}; B = {1, 2}	C = {1, 2, 3, 4, 5}	Упорядоченные массивы, причем второй не имеет уникальных элементов
A = {1, 2}; B = {1, 2, 3, 4, 5}	C = {1, 2, 3, 4, 5}	Упорядоченные массивы, причем первый не имеет уникальных элементов
A = {1, 2, 3, 4, 5}; B = {6, 7, 9}	C = {1, 2, 3, 4, 5, 6, 7, 9}	Упорядоченные массивы, у которых общих элементов нет
A = {1, 2}; B = {1, 2}	C = {1, 2}	Повторяющие друг друга массивы

Выполнение задания:

```
int getIncreasingJoinArray(const int *a, const int a_len, const
int *b, const int b_len, int *c, int *c_len) {
    *c_len = 0;
    int a_index = 0;
    int b_index = 0;
    while (a_index < a_len || b_index < b_len) {
        if (b_index >= b_len || a_index < a_len && a[a_index] <
b[b_index]) {
            c[*c_len] = a[a_index];
            a_index++;
        } else {
            c[*c_len] = b[b_index];
            if (a[a_index] == b[b_index]) {
                a_index++;
            }
            b_index++;
        }
        *c_len += 1;
    }
}
```

Задача 9:

Текст задания:

Даны массивы натуральных чисел A и B, упорядоченные по возрастанию. Получить упорядоченный по возрастанию массив C, содержащий все такие элементы, которые есть и в массиве A и в массиве B.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {1, 2, 3, 4, 5}; B = {1, 2, 6}	C = {1, 2}	Упорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {1, 2, 3, 4, 5}; B = {1, 2}	C = {1, 2}	Упорядоченные массивы, причем

B = {1, 2}		второй не имеет уникальных элементов
A = {1, 2}; B = {1, 2, 3, 4, 5}	C = {1, 2, 3, 4, 5}	Упорядоченные массивы, причем первый не имеет уникальных элементов
A = {1, 2, 3, 4, 5}; B = {6, 7, 9}	C = {}	Упорядоченные массивы, у которых общих элементов нет
A = {1, 2}; B = {1, 2}	C = {1, 2}	Повторяющие друг друга массивы

Выполнение задания:

```
int getIncreasingIntersectionArray(const int *a, const int a_len,
const int *b, const int b_len, int *c, int *c_len) {
    *c_len = 0;
    int a_index = 0;
    int b_index = 0;
    while (a_index < a_len || b_index < b_len) {
        if (b_index >= b_len || a_index < a_len && a[a_index] <
b[b_index]) {
            a_index++;
        } else {
            if (a[a_index] == b[b_index]) {
                c[*c_len] = a[a_index];
                a_index++;
                *c_len += 1;
            }
            b_index++;
        }
    }
}
```

Задача 10:

Текст задания:

Даны массивы натуральных чисел A и B, упорядоченные по возрастанию. Получить упорядоченный по возрастанию массив C, содержащий все элементы массива A, которых нет в B.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {1, 2, 3, 4, 5}; B = {1, 2, 6}	C = {3, 4, 5}	Упорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {1, 2, 3, 4, 5}; B = {1, 2}	C = {3, 4, 5}	Упорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {1, 2}	C = {}	Упорядоченные массивы, причем

B = {1, 2, 3, 4, 5}		массив А является подмассивом В
A = {1, 2, 3, 4, 5}; B = {6, 7, 9}	C = {1, 2, 3, 4, 5}	Упорядоченные массивы, у которых общих элементов нет
A = {1, 2}; B = {1, 2}	C = {}	Повторяющие друг друга массивы

Выполнение задания:

```
int getIncreasingExceptionArrayByA(const int *a, const int a_len,
const int *b, const int b_len, int *c, int *c_len) {
    *c_len = 0;
    int a_index = 0;
    int b_index = 0;
    while (a_index < a_len || a_index < a_len && b_index < b_len)
    {
        if (b_index >= b_len || a[a_index] < b[b_index]) {
            c[*c_len] = a[a_index];
            *c_len += 1;
            a_index++;
        } else {
            if (a[a_index] == b[b_index]) {
                a_index++;
            }
            b_index++;
        }
    }
}
```

Задача 11:

Текст задания:

Даны массивы натуральных чисел А и В, упорядоченные по возрастанию. Получить упорядоченный по возрастанию массив С, содержащий все элементы массива А, которых нет в В и все элементы массива В, которых нет в А.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {1, 2, 3, 4, 5}; B = {1, 2, 6}	C = {3, 4, 5, 6}	Упорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {1, 2, 3, 4, 5}; B = {1, 2}	C = {3, 4, 5}	Упорядоченные массивы, причем массив В является подмассивом А
A = {1, 2}; B = {1, 2, 3, 4, 5}	C = {3, 4, 5}	Упорядоченные массивы, причем массив А является подмассивом В
A = {1, 2, 3, 4, 5}; B = {6, 7, 9}	C = {1, 2, 3, 4, 5, 6, 7, 9}	Упорядоченные массивы, у которых общих элементов нет
A = {1, 2}; B = {1, 2}	C = {}	Повторяющие друг друга массивы

B = {1, 2}		
------------	--	--

Выполнение задания:

```
int getIncreasingExceptionArray(const int *a, const int a_len,
const int *b, const int b_len, int *c, int *c_len) {
    *c_len = 0;
    int a_index = 0;
    int b_index = 0;
    while (a_index < a_len || b_index < b_len) {
        if (b_index >= b_len || a_index < a_len && a[a_index] <
b[b_index]) {
            c[*c_len] = a[a_index];
            *c_len += 1;
            a_index++;
        } else {
            if (a[a_index] == b[b_index]) {
                a_index++;
            } else {
                c[*c_len] = b[b_index];
                *c_len += 1;
            }
            b_index++;
        }
    }
}
```

Задача 12:

Текст задания:

Даны массивы натуральных чисел A и B, упорядоченные по возрастанию. Определить, верно ли, что массив B содержит каждый элемент массива A.

Тестовые данные:

Входные данные:	Выходные данные:	Комментарий:
A = {1, 2, 3, 4, 5}; B = {1, 2, 6}	0	Упорядоченные массивы, в каждом из которых есть и уникальные, и повторяющиеся элементы
A = {1, 2, 3, 4, 5}; B = {1, 2}	0	Упорядоченные массивы, причем массив B является подмассивом A
A = {1, 2}; B = {1, 2, 3, 4, 5}	1	Упорядоченные массивы, причем массив A является подмассивом B
A = {1, 2, 3, 4, 5}; B = {6, 7, 9}	0	Упорядоченные массивы, у которых общих элементов нет
A = {1, 2}; B = {1, 2}	1	Повторяющие друг друга массивы

Выполнение задания:

```

int isIncreasingAPartOfB(const int *a, const int a_len, const int
*b, const int b_len) {
    int a_index = 0;
    int b_index = 0;
    int counter_of_doubles = 0;
    while (a_index < a_len || b_index < b_len) {
        if (b_index >= b_len || a_index < a_len && a[a_index] <
b[b_index]) {;
            a_index++;
        } else {
            if (a[a_index] == b[b_index]) {
                counter_of_doubles++;
                a_index++;
            }
            b_index++;
        }
    }
    return counter_of_doubles == a_len;
}

```

Функция main с вводом данных и выводом результатов работы функций с решением задачи:

```

int main() {
    //Общая часть, где осуществляется ввод сначала длины, а потом
элементов каждого из массивов A и B
    int first_len;
    scanf("%d\n", &first_len);
    int first_array[first_len];
    inputArray(first_array, first_len);

    int second_len;
    scanf("%d\n", &second_len);
    int second_array[second_len];
    inputArray(second_array, second_len);

    //Часть для задач, которые требуют создания массива C.
Изначальная длина устанавливается как сумма длин A и B, каждая из
функций меняет его в соответствии с количеством подходящих
элементов
    int result_len = first_len + second_len;
    int *result_array[result_len];

    //Вызовы функций для заполнения массива C и вывод массива C
    getJoinArray(first_array, first_len, second_array, second_len,

```

```

result_array, &result_len);
    printArray(result_array, result_len);

    getIntersectionArray(first_array, first_len, second_array,
second_len, result_array, &result_len);
    printArray(result_array, result_len);

    getExceptionArrayByA(first_array, first_len, second_array,
second_len, result_array, &result_len);
    printArray(result_array, result_len);

    getExceptionArray(first_array, first_len, second_array,
second_len, result_array, &result_len);
    printArray(result_array, result_len);

    getIncreasingJoinArray(first_array, first_len, second_array,
second_len, result_array, &result_len);
    printArray(result_array, result_len);

    getIncreasingIntersectionArray(first_array, first_len,
second_array, second_len, result_array, &result_len);
    printArray(result_array, result_len);

    getIncreasingExceptionArrayByA(first_array, first_len,
second_array, second_len, result_array, &result_len);
    printArray(result_array, result_len);

    getIncreasingExceptionArray(first_array, first_len,
second_array, second_len, result_array, &result_len);
    printArray(result_array, result_len);

    //Вывод значений истина-ложь (для задач 5, 6, 7, 12)

    printf("%d", isAPartOfB(first_array, first_len, second_array,
second_len));

    printf("%d", isArrayEqual(first_array, first_len,
second_array, second_len));

    printf("%d", isArrayDisjoint(first_array, first_len,
second_array, second_len));

    printf("%d", isIncreasingAPartOfB(first_array, first_len,
second_array, second_len));
}

```

