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/a: ст. группы ПВ-231

Чупахина София Александровна

Проверили:

Островский Алексей Мичеславович

Рязанов Юрий Дмитриевич

Белгород, 2023

[Библиотека «array\_input\_output»: 3](#__RefHeading___34)

[Задача 1: 3](#__RefHeading___35)

[Задача 2: 4](#__RefHeading___36)

[Задача 3: 5](#__RefHeading___37)

[Задача 4: 6](#__RefHeading___38)

[Задача 5: 7](#__RefHeading___39)

[Задача 6: 8](#__RefHeading___7)

[Задача 7: 9](#__RefHeading___40)

[Задача 8: 9](#__RefHeading___41)

[Задача 9: 10](#__RefHeading___42)

[Задача 10: 11](#__RefHeading___43)

[Задача 11: 12](#__RefHeading___44)

[Задача 12: 13](#__RefHeading___45)

[Функция main с вводом данных и выводом результатов работы функций с решением задачи: 14](#__RefHeading___46)

# Библиотека «array\_input\_output»:

#include <stdio.h>  
  
//осуществляет ввод массива a размера n  
void inputArray(int \*a, const size\_t n) {  
 for (size\_t i = 0; i < n; i++) {  
 scanf("%d", &a[i]);  
 }  
}  
//выводит массив a 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:

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

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

Получить массив С, содержащий все элементы массивов А и В 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 который 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:

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

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

Получить массив С, содержащий все такие элементы, которые есть и в массиве А и в

массиве В.

**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 который 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:

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

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

Получить массив С, содержащий все элементы массива А, которых нет в В.

**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:

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

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

Получить 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 = {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:

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

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

Определить, верно ли, что массив В содержит каждый элемент массива А.

**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:

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

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

Определить, верно ли, что массивы А и В 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 isArraysEqual(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:

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

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

Определить, верно ли, что в массивах А и В 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 А и В, упорядоченные по возрастанию.

Получить упорядоченный по возрастанию массив С, содержащий все элементы

массивов А и В.

**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 А и В, упорядоченные по возрастанию.

Получить упорядоченный по возрастанию массив С, содержащий все такие

элементы, которые есть и в массиве А и в массиве В.

**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 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 А и В, упорядоченные по возрастанию.

Получить упорядоченный по возрастанию массив С, содержащий все элементы

массива А, которых нет в В.

**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, 3, 4, 5} | C = {} | Упорядоченные массивы, причем массив A является подмассивом B |
| 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 B является подмассивом A |
| A = {1, 2};  B = {1, 2, 3, 4, 5} | C = {3, 4, 5} | Упорядоченные массивы, причем массив A является подмассивом B |
| 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 |

**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 А и В, упорядоченные по возрастанию.

Определить, верно ли, что массив В содержит каждый элемент массива А.

**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", isArraysEqual(first\_array, first\_len, second\_array, second\_len));

printf("%d", isArraysDisjoint(first\_array, first\_len, second\_array, second\_len));

printf("%d", isIncreasingAPartOfB(first\_array, first\_len, second\_array, second\_len));  
}