

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

“Московский государственный университет геодезии и картографии”

(МГИИГАИК)

Факультет геоинформатики и информационной безопасности

Кафедра геоинформационных систем и технологий

Лабораторная работа №4

“ Разработка консольного приложения с использованием функций”

Проверил:

ФИО

Выполнил:

Студент группы: 2024-ФГИИБ-ПИ-16

Бондаренко Максим Юрьевич

Цель работы

Разработать два консольных приложения для обработки данных массивов и множеств с использованием функций и указателей.

Задачи

1. Разработать приложение для обработки массивов:

- Найти максимальное значение в каждом массиве и вычесть его из всех элементов массивов.
- Вывести исходные и преобразованные массивы.

2. Разработать приложение для обработки множеств:

- Найти наименьший элемент, который не принадлежит множеству A, но есть в множестве B.

Задание 1: Обработка массивов

Формулировка:

Заданы два массива A(5) и B(5). Найти наибольшее значение в каждом массиве и вычесть его из всех элементов массивов. На печать вывести исходные и преобразованные массивы.

Исходный код и проект находятся в репозитории на GitHub:

https://github.com/Maxrgrz/info_Cplus

Формулировка:

Заданы два массива A(5) и B(5). Найти наибольшее значение в каждом массиве и вычесть его из всех элементов массивов. На печать вывести исходные и преобразованные массивы.

Листинг 1: Код программы

```
#include <iostream>
#include <algorithm> // Для функции max_element

// Функция для поиска максимального элемента массива
void findMaxElement(const int* arr, int size, int& maxVal) {
    maxVal = *std::max_element(arr, arr + size);
}
```

```

// Функция для преобразования массива путем вычитания максимального значения
void transformArray(int* arr, int size, int maxVal) {
    for (int i = 0; i < size; i++) {
        arr[i] -= maxVal;
    }
}

// Функция для вывода массива на экран
void printArray(const int* arr, int size) {
    for (int i = 0; i < size; i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}

int main() {
    const int N = 5; // Размер массивов
    int A[N], B[N]; // Два массива

    // Ввод элементов массива A
    std::cout << "Введите 5 элементов массива A: ";
    for (int i = 0; i < N; i++) {
        std::cin >> A[i];
    }

    // Ввод элементов массива B
    std::cout << "Введите 5 элементов массива B: ";
    for (int i = 0; i < N; i++) {
        std::cin >> B[i];
    }

    // Исходные массивы
    std::cout << "Исходный массив A: ";
    printArray(A, N);
    std::cout << "Исходный массив B: ";
    printArray(B, N);

    // Поиск максимального значения и преобразование массива A
    int maxA, maxB;
    findMaxElement(A, N, maxA);
    findMaxElement(B, N, maxB);

    transformArray(A, N, maxA);
    transformArray(B, N, maxB);
}

```

```

// Преобразованные массивы
std::cout << "Преобразованный массив A: ";
printArray(A, N);
std::cout << "Преобразованный массив B: ";
printArray(B, N);

return 0;
}

```

Описание программы

1. Функция findMaxElement:

- Принимает массив по указателю `const int* arr` и находит максимальное значение в нем.
- Результат возвращается через ссылку на переменную `maxVal`.

2. Функция transformArray:

- Выполняет вычитание максимального значения из всех элементов массива.
- Принимает массив по указателю `int* arr`.

3. Функция printArray:

- Печатает элементы массива.

4. Основной блок программы:

- Считывает массивы A и B размером 5.
- Находит их максимальные значения и выполняет преобразование.
- Выводит исходные и преобразованные массивы.

Тестирование программы:

Входные данные	Исходный A	Исходный B	Преобразованный A	Преобразованный B
5 7 2 8 1	5 7 2 8 1	3 9 4 1 6	-3 -1 -6 0 -7	-6 0 -5 -8 -3
10 15 20 5 30	10 15 20 5 30	40 25 30 10 5	-20 -15 -10 -25 0	0 -15 -10 -30 -35
1 1 1 1 1	1 1 1 1 1	2 2 2 2 2	0 0 0 0 0	0 0 0 0 0

Задание 2: Обработка множеств

Формулировка:

Заданы два множества A и B. Найти наименьший элемент, который не принадлежит множеству A, но есть в множестве B.

Листинг 2: Код программы

```
#include <iostream>

#include <set>

#include <limits>

// Функция для нахождения наименьшего элемента, который есть в B, но
// отсутствует в A
int findMinExclusiveElement(const std::set<int>& A, const std::set<int>& B) {
    for (const int& elem : B) {
        if (A.find(elem) == A.end()) { // Элемент не найден в A
            return elem;
        }
    }
    return std::numeric_limits<int>::max(); // Если нет такого элемента
}

int main() {
    std::set<int> A, B;
    int sizeA, sizeB, element;
```

```
// Ввод множества A

std::cout << "Введите количество элементов множества A: ";
std::cin >> sizeA;


std::cout << "Введите элементы множества A: ";
for (int i = 0; i < sizeA; i++) {
    std::cin >> element;
    A.insert(element);
}


// Ввод множества B

std::cout << "Введите количество элементов множества B: ";
std::cin >> sizeB;
std::cout << "Введите элементы множества B: ";
for (int i = 0; i < sizeB; i++) {
    std::cin >> element;
    B.insert(element);
}


// Поиск и вывод наименьшего элемента
int result = findMinExclusiveElement(A, B);
if (result != std::numeric_limits<int>::max()) {
    std::cout << "Наименьший элемент, который есть в B, но отсутствует в A: " <<
result << std::endl;
} else {
    std::cout << "Нет элементов в B, которые отсутствуют в A." << std::endl;
}
```

```
    return 0;  
}
```

Тестирование программы:

Входные данные А	Входные данные В	Наименьший элемент
1 2 3	3 4 5	4
10 20 30	40 50 60	40
1 1 1	1 2 3	2

Вывод

В ходе работы были разработаны два консольных приложения для обработки массивов и множеств с использованием функций и указателей. Проведено тестирование, подтвердившее корректность работы программ.