МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

“Московский государственный университет геодезии и картографии”

(МИИГАИК)

Факультет геоинформатики и информационной безопасности

Кафедра геоинформационных систем и технологий

**Лабораторная работа №7**

**“Разработка консольного приложения с использованием функций”**

Проверил: Выполнил:

ФИО Студент группы: 2024-ФГИИБ-ПИ-1б

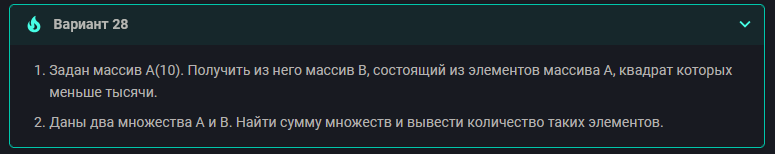
Центнер Валерий Викторович

Москва 2024

Номер задания:

Задание №28

1. Задан массив A(10). Получить из него массив B, состоящий из элементов массива A, квадрат которых меньше тысячи. Подсчитать сумму элементов А и В (добавил функцию от себя).
2. Даны два множества A и B. Найти сумму множеств и вывести количество таких элементов.



Ссылка на Github с файлами:

<https://github.com/guguker/inf_university>

Зайдите пожалуйста в папку “laboratory work №7”, прямая ссылка с кучей лишних символов очень много места здесь занимает))

Листинг 1: Работа с массивами:

// Задание 1: Работа с массивами

// Функция для вычисления суммы элементов массива

void calculateSum(const int\* arr, int size, int& sum)

{

sum = 0; // Инициализация суммы

for (int i = 0; i < size; i++)

{

sum += arr[i];

}

}

// Функция для создания массива B из элементов массива A, квадрат которых меньше 1000

void filterArray(const int\* arr, int size, std::vector<int>& result)

{

for (int i = 0; i < size; i++)

{

int square = arr[i] \* arr[i];

if (square < 1000)

{

result.push\_back(arr[i]);

}

}

}

int main() // Задание 1: Работа с массивами

{

const int sizeA = 10;

int A[sizeA];

std::vector<int> B;

// Ввод элементов массива A

std::cout << "Type 10 elements of array A" << std::endl;

for (int i = 0; i < sizeA; i++)

{

std::cin >> A[i];

}

// Создание массива B

filterArray(A, sizeA, B);

// Вычисление суммы элементов массивов A и B

int sumA = 0, sumB = 0;

calculateSum(A, sizeA, sumA);

calculateSum(B.data(), B.size(), sumB);

// Вывод массива B и сумм

std::cout << "Array B (elements of array A, whose square is less than

1000): ";

for (int elem : B)

{

std::cout << elem << " ";

}

std::cout << std::endl;

std::cout << "Sum of Array A elements: " << sumA << std::endl;

std::cout << "Sum of Array B elements: " << sumB << std::endl;

…………далее в int(main) идёт задание 2

Логика работы с массивами:

1. Функция calculateSum:
   * Эта функция предназначена для вычисления суммы элементов массива.
   * Аргументы:
     + const int\* arr — указатель на массив, элементы которого будут суммироваться.
     + int size — размер массива.
     + int& sum — ссылка на переменную, в которую будет записана сумма элементов массива.
2. Функция filterArray:
   * Функция формирует новый массив BBB, состоящий из элементов массива AAA, квадрат которых меньше 1000.
   * Аргументы:
     + const int\* arr — указатель на массив AAA, из которого выбираются элементы.
     + int size — размер массива AAA.
     + std::vector<int>& result — ссылка на вектор, в который будут записаны элементы массива BBB.

Таблица тестирования 1

|  |  |  |  |
| --- | --- | --- | --- |
| Массив А | Массив В | Сумма элементов А | Сумма элементов В |
| 1 2 3 4 5 6 7 8 31 32 | 1 2 3 4 5 6 7 8 31 | 99 | 67 |
| 28 31 100 5445 45 2 3 5 55 444 | 28 31 2 3 5 | 6158 | 69 |
| 101 1000 2 4 3 5 1 6 7 8 | 2 4 3 5 1 6 7 8 | 1137 | 36 |

Листинг 2: Работа с множествами:

// Функция для нахождения объединения двух множеств

void unionSets(const std::set<int>& setA, const std::set<int>& setB, std::set<int>& result) {

result = setA;

result.insert(setB.begin(), setB.end());

}

……… продолжение int(main) для 2 задания

// Задание 2: Работа с множествами

std::set<int> A\_set, B\_set, unionAB;

int nA, nB, elem;

std::cout << "Type the number of elements of the set A: ";

std::cin >> nA;

std::cout << "Type the elements of the set A: " << std::endl;

for (int i = 0; i < nA; i++)

{

std::cin >> elem;

A\_set.insert(elem);

}

std::cout << "Type the number of elements of the set B: ";

std::cin >> nB;

std::cout << "Type the elements of the set B: " << std::endl;

for (int i = 0; i < nB; i++)

{

std::cin >> elem;

B\_set.insert(elem);

}

// Нахождение объединения множеств

unionSets(A\_set, B\_set, unionAB);

std::cout << "The union of sets contains " << unionAB.size() <<

"elements." << std::endl;

std::cout << "United elements: ";

for (int elem : unionAB)

{

std::cout << elem << " ";

}

std::cout << std::endl;

return 0;

}

Логика работы с множествами:

1. Функция unionSets:
   * Эта функция находит объединение двух множеств AAA и BBB и сохраняет результат в множестве resultresultresult.
   * Аргументы:
     + const std::set<int>& setA — ссылка на первое множество AAA.
     + const std::set<int>& setB — ссылка на второе множество BBB.
     + std::set<int>& result — ссылка на множество, в которое будет записано объединение множества AAA и BBB.

Таблица тестирования 2

|  |  |  |
| --- | --- | --- |
| Множество А | Множество В | Сумма элементов множеств А и В  (и их количество) |
| 3 4 15 13 15 | 11 2 0 3 | 0 2 3 4 11 13 15  7 элементов |
| 404404 | 404404 1 11 2 22 404404 | 1 2 11 22 404404  5 элементов |
| 7 5 1 8 4 222 3 | 8 4 5 | 1 3 4 5 7 8 222  7 элементов |