**Что сделал:**

1. Добавил пояснение перед проверкой
2. Указал, зачем нужна эта проверка

**Итоговый код:**

cpp

Copy

Download

*// Проверка необходима для корректной работы алгоритма,*

*// так как при отрицательных значениях требуется особая обработка*

if (x > 0) {

*// ... остальной код ...*

}

**Команды Git:**

bash

Copy

Download

git add laba10.1.cpp

git commit -m "Добавил поясняющий комментарий к условию"

**Что получилось в итоге:**

1. Код стал читаемее
2. Логика работы понятнее благодаря комментариям
3. Добавлена проверка входных данных
4. Все изменения зафиксированы отдельными коммитами

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

#include <math.h>

// Объявление переменных (каждая с новой строки)

int A[20]; // Основной рабочий массив

int k1; // Первый индекс для обработки

int k2; // Второй индекс для обработки

FILE\* file; // Указатель на файл для записи

FILE\* readFile; // Указатель на файл для чтения

int sum; // Переменная для хранения суммы

void initializeArray(int\* arr, int size) {

srand(time(0));

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

arr[i] = rand() % 21 - 10;

}

}

void printArrayToFile(const int\* arr, int size, FILE\* file) {

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

fprintf(file, "%d ", arr[i]);

}

fprintf(file, "\n");

}

void processArray(int\* arr, int size, int k1, int k2) {

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

if (arr[i] > 0) {

arr[i] -= arr[k1];

}

else {

arr[i] -= arr[k2];

}

}

}

int sumArray(int\* arr, int size) {

if (size == 0) {

return 0;

}

return arr[size - 1] + sumArray(arr, size - 1);

}

void readArrayFromFile(int\* arr, int size, FILE\* file) {

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

fscanf(file, "%d", &arr[i]);

}

}

int main() {

setlocale(LC\_ALL, "");

// Проверка на положительность первого элемента массива

initializeArray(A, 20);

if (A[0] > 0) {

// Условное выражение проверяет знак первого элемента

printf("Первый элемент массива положительный: %d\n", A[0]);

}

else {

printf("Первый элемент массива не положительный: %d\n", A[0]);

}

// Открытие файла для добавления данных

file = fopen("E:/Programmirovanie/labworks/lab10/results.txt", "a");

if (file == NULL) {

printf("Ошибка открытия файла!\n");

return 1;

}

// Генерация случайных индексов

k1 = rand() % 20;

k2 = rand() % 20;

// Запись исходного массива в файл

fprintf(file, "Исходный массив A:\n");

printArrayToFile(A, 20, file);

// Обработка массива

processArray(A, 20, k1, k2);

fprintf(file, "Обработанный массив A:\n");

printArrayToFile(A, 20, file);

fprintf(file, "k1 = %d, k2 = %d\n", k1, k2);

sum = sumArray(A, 20);

fprintf(file, "Сумма A: %d\n", sum);

// Закрытие файла

fclose(file);

printf("Результаты сохранены в файл results.txt\n");

// Чтение данных из файла

readFile = fopen("results.txt", "r");

if (readFile == NULL) {

printf("Ошибка открытия файла для чтения!\n");

return 1;

}

readArrayFromFile(A, 20, readFile);

// Обработка прочитанного массива

processArray(A, 20, k1, k2);

printf("Обработанный массив из файла: ");

for (int i = 0; i < 20; i++) {

printf("%d ", A[i]);

}

printf("\n");

fclose(readFile);

return 0;

}

