Лабораторная работа №8

Бинарные файлы.

ЗАДАНИЕ

Вариант 6

1. Создать файл, содержащий целые числа. Подсчитать сумму положительных элементов файла.
2. Создать файл f, содержащий одномерный массив из 10 вещественных чисел. Записать в файл g все отрицательные числа из этого массива.

ХОД РАБОТЫ

1. Создать файл, содержащий целые числа. Подсчитать сумму положительных элементов файла.

Результат работы программы представлен в соответствии с рисунком 1.

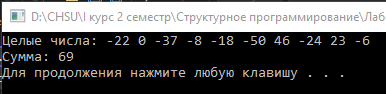


Рисунок 1 – результаты работы программы.

Текст программы:

#include <iostream>

#include <fstream>

#include <time.h>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "rus");

const int n = 10;

ofstream wfile("input.bin", ios::binary | ios::out);

srand(time(0));

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

int num = rand() % 100 - 50;

wfile.write((const char\*)&num, sizeof(num));

}

wfile.close();

ifstream rFile("input.bin", ios::binary | ios::in);

if (!rFile) {

return -1;

}

int x, s = 0;

cout << "Целые числа: ";

while (rFile.read((char\*)&x, sizeof(x))) {

if (x > 0)

s += x;

cout << x << " ";

}

cout << endl << "Сумма: " << s << endl;

rFile.close();

system("pause");

return 0;

}

Алгоритм программы:

Функция write() выполняет запись переменной num в файл как последовательность байтов типа const char.

Функция write() выполняет запись данных переменной num в файл.

Значение переменной num считывается как последовательность байтов, представленных как массив символов типа const char, и записывается в файл. Количество байтов для записи определяется с помощью sizeof(num), чтобы учесть размер переменной num.

Подключаем библиотеки time.h и fstream. Объявляем целочисленную константу n, принимающую значение 10.

С помощью класса ofstream (для записи) открываем бинарный файл «input.bin». В цикле от 0 до n задаём случайное значение целочисленной переменной num и записываем его в бинарный файл. После выхода из цикла закрываем файл.

С помощью класса ifstream (для записи) открываем бинарный файл «input.bin». Объявляем целочисленные переменные x и s (значением которой является 0). Функция read извлекает символы из потока и сохраняет их по адресу указателя char, переданного ей в качестве первого аргумента. Если значение x больше 0 (число положительное), то суммируем значение в переменной s. Когда функция извлечёт все символы из потока, закрываем файл и выводим значение переменной s (сумму положительных целых чисел).

1. Что это за функция?

- Это вызов метода write() для объекта file, где file является экземпляром класса ofstream. Метод write() предоставляет возможность записи данных в файл.

2. Что она делает?

- Функция записывает значение переменной num в файл. Значение переменной num интерпретируется как последовательность байтов типа const char, которые затем записываются в файл.

3. Для чего нужна?

- Функция нужна для сохранения данных переменной num в файле. Это может быть полезно, к примеру, для сохранения состояния программы или для обмена данными между программами.

4. Как она работает?

- Функция сначала преобразует указатель на num в указатель на тип const char, чтобы интерпретировать значение переменной как последовательность байтов. Затем, с помощью метода write() экземпляра класса ofstream, эта последовательность байтов записывается в файл.

5. Зачем мы используем (const char\*) и &num?

- Мы используем (const char\*), чтобы произвести явное преобразование типа указателя на переменную num к типу const char\*. Это позволяет рассматривать значение переменной num как последовательность байтов для записи в файл.

- Мы используем &num, чтобы получить указатель на адрес памяти, где хранится значение переменной num. Это необходимо, чтобы передать этот адрес в write() в качестве указателя на данные, которые нужно записать.

6. Есть ли особенности в работе с типом const char\* и write()?

- const char\* часто используется для работы с последовательностями символов, такими как строки в C++. Он позволяет рассматривать данные как массив символов.

- Метод write() имеет второй параметр streamsize, который определяет количество байтов, которые нужно записать. В нашем случае, мы используем sizeof(num) для определения размера переменной num в байтах, чтобы гарантировать, что все данные будут записаны.

1. Создать файл f, содержащий одномерный массив из 10 вещественных чисел. Записать в файл g все отрицательные числа из этого массива.

Результат работы программы представлен в соответствии с рисунком 1.

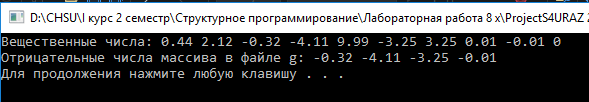


Рисунок 1 – результаты работы программы.

Текст программы:

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "rus");

double a[] = { 0.44,2.12,-0.32,-4.11,9.99,-3.25,3.25,0.01,-0.01,0.00 };

FILE\* file = fopen("double.bin", "wb");

fwrite(&a, sizeof(double), 10, file);

fclose(file);

double b[10];

FILE\* fileo = fopen("double.bin", "rb");

FILE\* gfile = fopen("g.bin", "wb");

fread(&b, sizeof(double), 10, fileo);

for (int i = 0; i < 10; i++) {

if (b[i] < 0) {

fwrite(&b[i], sizeof(double), 1, gfile);

}

}

fclose(fileo);

fclose(gfile);

/\*double c[10];

FILE\* fileend = fopen("g.bin", "rb");

fread(&c, sizeof(double), 10, fileend);

for (int i = 0; i < ftell(fileend) / sizeof(double); i++){

cout << c[i] << " ";

}

fclose(fileend); \*/

system("pause");

return 0;

}

Алгоритм программы:

Объявляем вещественный одномерный массив a из 10 элементов и каждому присваивается вещественное значение.

Создаем файловый объект file с помощью вызова функции fopen с параметрами «double.txt» (имя файла) и «wb» (режим доступа к файлу – для записи бинарных файлов), которая возвращает указатель на объект файла. В бинарный файл «double.bin» записываются вещественные значения элементов одномерного массива а. Закрываем файл.

Объявляем вещественный одномерный массив b из 10 элементов. Создаём файловые объекты fileo и gfile для чтения и записи бинарных файлов. Считываем данные файла «double.bin» и заносим значения в массив b. В цикле от 0 до 10 сравниваем значение каждого элемента массива с 0. Если меньше, то это отрицательное число, поэтому записываем значение элемента в файл «g.bin». Закрываем файлы.

Вывод:

В ходе выполнения лабораторной работы мы научились использовать рекурсивные функции и закрепили полученные навыки.