МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Расчетно-графическая работа

по дисциплине «Программирование»

Работа с памятью на низком уровне.

Структуры данных в двоичном коде.

Группа: АВТ-041

Студент: Антонов А.

Преподаватель: Балакин В.

НОВОСИБИРСК 2021

**Задача 1 тема 1 (3 вариант)**

Упакованный массив целых переменных. Байт-счетчик, имеющий положительное значение n, предваряет последовательность из n различных целых переменных, байт-счетчик, имеющий отрицательное значение -n, обозначает n подряд идущих одинаковых значений целой переменной. Пример:

-исходная последовательность: **2 3 3 3 5 2 4 4 4 4 4 8 -6 8**

-упакованная последовательность: **(1) 2 (-3) 3 (2) 5 2 (-5) 4 (3) 8 -6 8**

**Проектирование программы**

Идея: функция, в которой за счёт сравнения значений и их сортировки упаковывается массив.

Составные части программы:

1. Вводим последовательность:

while (cin >> a) { // вводим массив

v.push\_back(a);

}

v.push\_back(INT\_MIN);

1. Цикл, в котором при нахождении рядом стоящих одинаковых значениях учитывается и одинаковое значение которое стоит не рядом:

for (int i = 0; i < v.size() - 1; ++i) {

if (v[i] == v[i + 1]) { // если соседние числа равны

int counter = 0; // счётчик одинаковых чисел

for (int j = i; j < v.size() - 1; ++j) {

if (v[j] == v[i]) { // поиск кол-ва всех одинаковых значений

++counter; // закрывается счётчик

}

}

1. Вывод полученного значения, и учёт того на каком значении мы остановились:

cout << '(' << -counter << ')' << ' ' << v[i] << ' '; // запись кол-ва цифры перед самой цифрой с "-" в начале

i += counter - 1;

1. Если число одинокое, то нумерация таких значений начинается с 1 и смотрится не равно ли это значение предыдущим:

else {

int counter = 1;

for (int j = i + 1; j < v.size() - 1; ++j) {

if (v[j] != v[j - 1]) { // если не равен предыдущему то счетчик увелич. 1 раз (просто позиция единичной цифры 1,2,3,...)

++counter;

}

else {

--counter;

break;

}

}

1. Вывод получившегося значения:

cout << '(' << counter << ')' << ' ';

for (int j = i; j < i + counter && j < v.size() - 1; ++j) {

cout << v[j] << ' '; // вывод этой единичной цифры с номером перед

}

i += counter - 1; // обнуление счетчика

1. Функция main:

int main() {

input();

package(v);

}

Переменные:

v[] - массив

a - значение последовательности

i, j, counter - переменные-счётчик

**Текст программы с комментариями:**

#include<vector>

#include<iostream>

#include<climits>

using namespace std;

vector<int> v;

void input() {

int a;

while (cin >> a) { // вводим массив

v.push\_back(a);

}

v.push\_back(INT\_MIN);

}

void package(const vector<int>& v) { // упаковка массива

for (int i = 0; i < v.size() - 1; ++i) {

if (v[i] == v[i + 1]) { // если соседние числа равны

int counter = 0; // счётчик одинаковых чисел

for (int j = i; j < v.size() - 1; ++j) {

if (v[j] == v[i]) { // поиск кол-ва всех одинаковых значений

++counter; // закрывается счётчик

}

}

cout << '(' << -counter << ')' << ' ' << v[i] << ' '; // запись кол-ва цифры перед самой цифрой с "-" в начале

i += counter - 1;

}

else {

int counter = 1;

for (int j = i + 1; j < v.size() - 1; ++j) {

if (v[j] != v[j - 1]) { // если не равен предыдущему то счетчик увелич. 1 раз (просто позиция единичной цифры 1,2,3,...)

++counter;

}

else {

--counter;

break;

}

}

cout << '(' << counter << ')' << ' ';

for (int j = i; j < i + counter && j < v.size() - 1; ++j) {

cout << v[j] << ' '; // вывод этой единичной цифры с номером перед

}

i += counter - 1; // обнуление счетчика

}

}

return;

}

int main() {

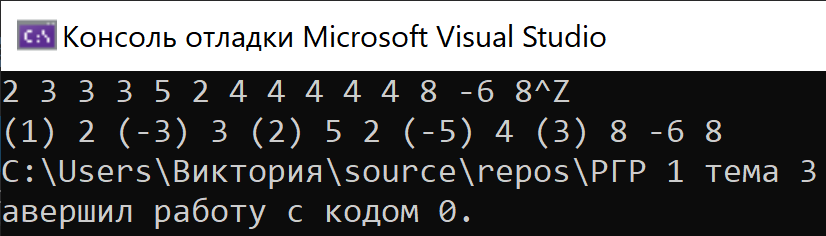
input();

package(v);

}

**Пример работы программы:**

**Входные данные: 2 3 3 3 5 2 4 4 4 4 4 8 -6 8**



**Вывод:** Программа оптимально выполняет поставленную задачу.

**Задача 2 тема 4 (11 вариант)**

В файле записей фиксированной длины содержится двоичное дерево. Вершина содержит значение типа int, а также номера соответствующих записей для правого и левого потомков. Реализовать функцию включения нового значения в существующий файл в виде новой вершины двоичного дерева.

**Проектирование программы**

Идея: Создание двух функций save и show. В функции save учитывается структура и расположение в ней значений для включения нового. Функция show для вывода структуры на консоль.

Составные части программы:

1. Создание структуры:

struct node

{

int value; // новая вершина

int left; // левый указатель

int right; // правый указатель

};

1. Открытие и создание файла для чтения/записи, и повторение этого через goto:

if (fopen\_s(&fp, filename, "r+b") != 0) { fopen\_s(&fp, filename, "w+b"); goto z2; }

rewind(fp); // перемещает указатель на начало потока файла

1. Контролируется положение значений и их включение в структуру за счёт pos и pos2:

fread(&t, sizeof(struct node), 1, fp); // считывание файла

if (feof(fp)) goto z2; // контролируется положение указателя в файле

if (t.left == 0 || t.right == 0) // если левая или правая ветка равна 0

{

pos = ftell(fp); // значение соответствующее количеству байт потока от начала файла

fseek(fp, 0, 2); // устанавление указателя в файле для ввода вывода значения

pos2 = ftell(fp);

fseek(fp, pos - sizeof(struct node), 0); // добавление значения с учетом заполнения в других нодах и смещения

pos2 = pos2 / sizeof(struct node) + 1;

if (t.left == 0) t.left = pos2; else t.right = pos2; // заполнение одной из нод

fwrite(&t, sizeof(struct node), 1, fp); // запись в файл

goto z2;

}

goto z;

1. Запись полученного в файл:

z2:;

fseek(fp, 0, 2);

fwrite(&p, sizeof(struct node), 1, fp);

fclose(fp);

}

1. Функция show для вывода структуры на консоль:

void show(char\* filename) // функция вывода содержания файла на консоль

{

FILE\* fp;

struct node t;

fopen\_s(&fp, filename, "rb");

rewind(fp);

zz:;

fread(&t, sizeof(struct node), 1, fp);

if (feof(fp)) goto zz2;

std::cout << t.value << " " << t.left << " " << t.right << "\n";

goto zz;

zz2:;

fclose(fp);

}

1. Функция main:

int main()

{

char filename[] = "tree.dat";

save(filename, 1);

show(filename);

system("pause");

return 0;

}

Переменные:

node - структура

z2, z, zz2, zz, z: - лейблы

value - новая вершина

left, right - левый и правый указатель

fp - указатель на файл

pos, pos2 - значение соотв. кол-ву байт для устанавления указателя в файле и заполнения ноды

t, p - левая и правая ветка

**Текст программы с комментариями:**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

struct node

{

int value; // новая вершина

int left; // левый указатель

int right; // правый указатель

};

void save(char\* filename, int v) // функция включения нового значения

{ // запись структуры в файл

FILE\* fp;

int pos, pos2;

struct node t, p = { v, 0, 0 };

if (fopen\_s(&fp, filename, "r+b") != 0) { fopen\_s(&fp, filename, "w+b"); goto z2; }

rewind(fp); // перемещает указатель на начало потока файла

z:;

fread(&t, sizeof(struct node), 1, fp); // считывание файла

if (feof(fp)) goto z2; // контролируется положение указателя в файле

if (t.left == 0 || t.right == 0) // если левая или правая ветка равна 0

{

pos = ftell(fp); // значение соответствующее количеству байт потока от начала файла

fseek(fp, 0, 2); // устанавление указателя в файле для ввода вывода значения

pos2 = ftell(fp);

fseek(fp, pos - sizeof(struct node), 0); // добавление значения с учетом заполнения в других нодах и смещения

pos2 = pos2 / sizeof(struct node) + 1;

if (t.left == 0) t.left = pos2; else t.right = pos2; // заполнение одной из нод

fwrite(&t, sizeof(struct node), 1, fp); // запись в файл

goto z2;

}

goto z;

z2:;

fseek(fp, 0, 2);

fwrite(&p, sizeof(struct node), 1, fp);

fclose(fp);

}

void show(char\* filename) // функция вывода содержания файла на консоль

{

FILE\* fp;

struct node t;

fopen\_s(&fp, filename, "rb");

rewind(fp);

zz:;

fread(&t, sizeof(struct node), 1, fp);

if (feof(fp)) goto zz2;

std::cout << t.value << " " << t.left << " " << t.right << "\n";

goto zz;

zz2:;

fclose(fp);

}

int main()

{

char filename[] = "tree.dat";

save(filename, 1);

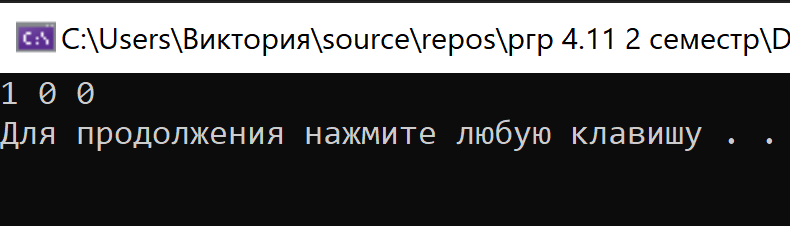
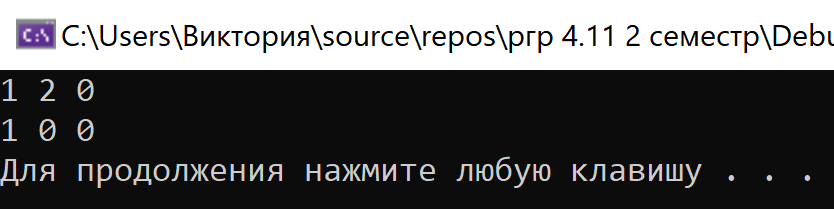
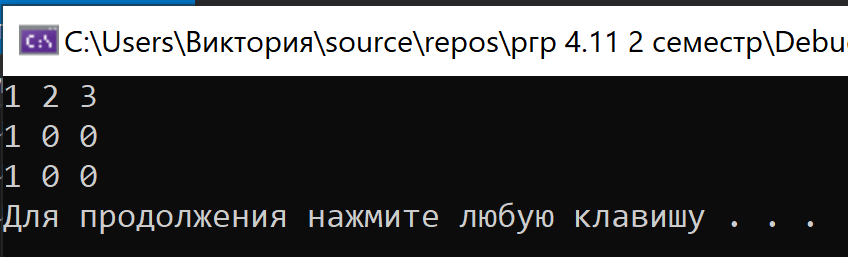
show(filename);

system("pause");

return 0;

}

**Пример работы программы:**

1. 
2. 
3.  и т.д.

**Вывод:** программа работает корректно.