МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Методы сортировки

ОТЧЕТ

по лабораторной работе № 5

по дисциплине

Технологии программирования \_

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Капранов С.Н.. (подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Антонова С.И.

(подпись) (фамилия, и.,о.)

18-ИСТ-2\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород  
2019

Содержание

[**Текст задания** 3](#_Toc23886006)

# **Текст задания**

**Задача 1.** Имеется N камней веса А1,А2,...,АN.

Необходимо разбить их на две кучи таким образом, чтобы веса куч отличались не более чем в 2 раза. Если этого сделать нельзя, то указать это.

**Текст программы с комментариями**

#include <iostream>

#include <stdlib.h>

using namespace std;

struct pile

{

int weight;

pile \*next;

};

void Del(pile\* del);//удаление очереди

//принимает указатель на начало

int Protect(); //проверка ввода

//возвращает разрешенное значение

int main()

{

setlocale(LC\_ALL, "Rus"); //подключаем русскую локализацию

int i, j; //счетчики

int quan = 0; //количество камней

int buf; //промежуточная переменная для перестановок

float sum1 = 0, sum2 = 0; //веса куч

pile \*pile1=NULL, \*cur1; //указатели на первый и текущий элементы первой кучи

pile \*pile2=NULL, \*cur2; //указатели на первый и текущий элементы второй кучи

char answer;

for(;;) //в качестве интерфейса

{

pile1=NULL;

pile2=NULL;

sum1 = 0;

sum2 = 0;

cout<<"Сколько камней?"<<endl;

quan = Protect();

int \*rocks = new int[quan]; //создание массива весов

cout<<"Введите веса камней"<<endl;

for (i = 0; i < quan; i++)

{

rocks[i] = Protect();

}

for (int j = quan-1; j > 0; j--) //сортировка по убыванию

{

for (i = j; i >= 0; i--)

{

if (rocks[(i-1)/2] > rocks[i])

{

buf = rocks[i];

rocks[i] = rocks[(i-1)/2];

rocks[(i-1)/2] = buf;

}

}

buf = rocks[0];

rocks[0] = rocks[j];

rocks[j] = buf;

}

for (i = 0; i < quan; i++) //разбиение камней на 2 кучи

{

if(sum1 <= sum2) //добавляем камню в ту кучу, вес которой меньше

{

if(pile1 == NULL)

{

pile1 = new pile;

cur1 = pile1;

}

else

{

cur1->next = new pile;

cur1 = cur1->next;

}

cur1->weight = rocks[i];

sum1 = sum1 + rocks[i];

}

else

{

if(pile2 == NULL)

{

pile2 = new pile;

cur2 = pile2;

}

else

{

cur2->next = new pile;

cur2 = cur2->next;

}

cur2->weight = rocks[i];

sum2 = sum2 + rocks[i];

}

}

if(pile1) cur1->next = NULL;

if(pile2) cur2->next = NULL;

delete[] rocks;

if((sum1 == 0)||( sum2 == 0))

{

cout<<"Невозможно";

}

else if(((sum1/sum2) > 2)||((sum2/sum1) > 2))

{

cout<<"Невозможно";

}

else

{

cur1 = pile1;

cout<<endl;

while(cur1)

{

cout<<cur1->weight<<' ';

cur1 = cur1->next;

}

cout<<endl;

cur2 = pile2;

while(cur2)

{

cout<<cur2->weight<<' ';

cur2 = cur2->next;

}

}

if(pile1) Del(pile1);

if(pile2) Del(pile2);

cout<<endl<<"Если желаете продолжить, введите 'y'. Для закрытия программы введите любой другой символ"<<endl;

cin>>answer;

if(answer != 'y')break;

system("cls");

}

}

int Protect()

{

int x;

for(;;)

{

cin>>x;

if (cin.fail()==true) //если встретилась ошибка

{

cin.clear(); //восстановление потока

}

for(;;) //читаем лишние символы

{

if(cin.get()=='\n')break;

}

if(x > 0) break; //выходим из цикла, если введено положительное число

cout<<"Пожалуйста, вводите только положительные числа"<<endl;

}

return x;

}

void Del(pile\* cur)

{

pile\* del;

while (cur)

{

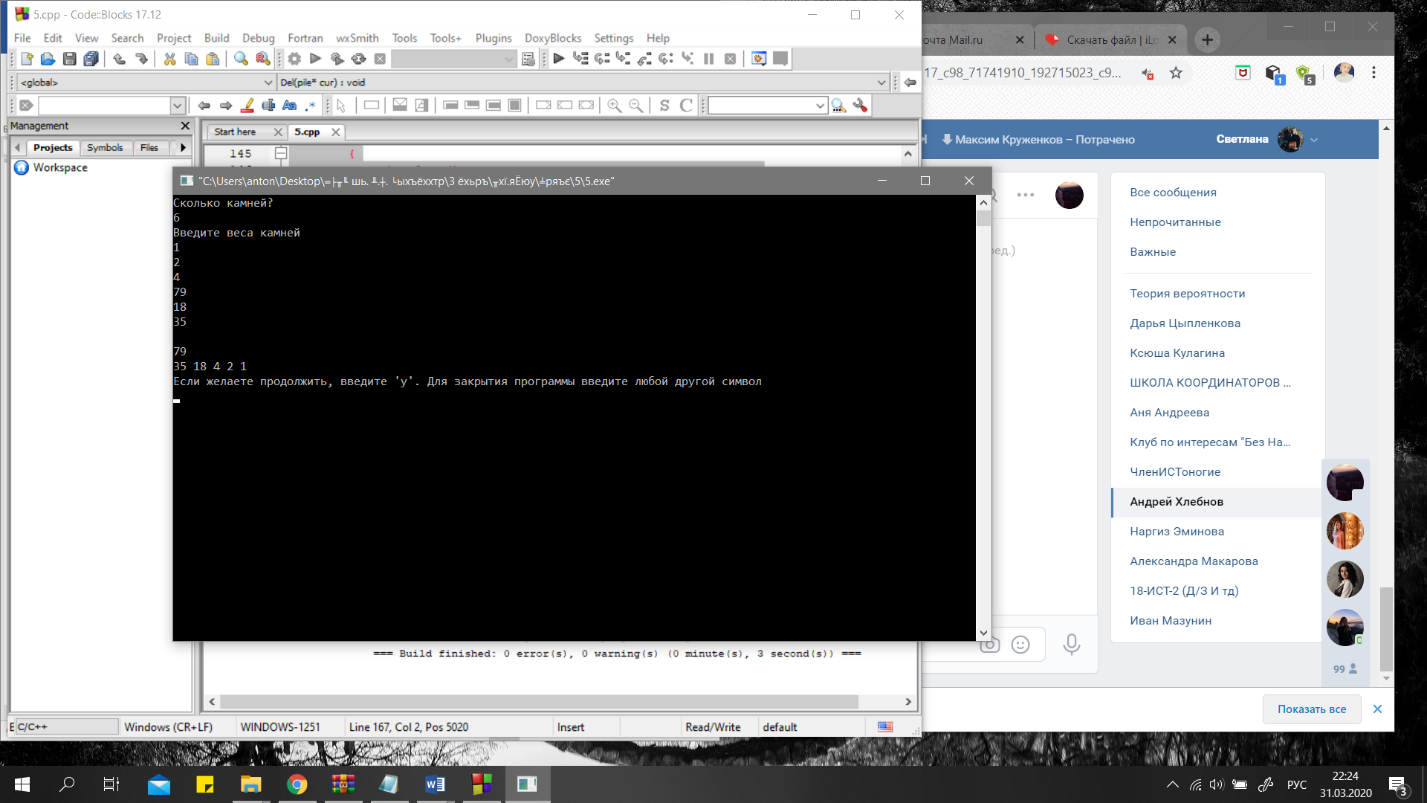
del = cur;

cur = cur->next;

delete del;

}

}

**Принтскрины экранов с входными и выходными данными.**