**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе №10**

**РАБОТА С БИНАРНЫМИ ФАЙЛАМИ**

**По курсу «Алгоритмизация и программирование»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Кармаев Александр Андреевич | БПМ211 | 15.10.2021 |
|  |
|  |
|  |

**Москва – 2021 г.**

**ЗАДАНИЕ (вариант №13)**

*Программа должна быть разбита на несколько функций и обязательно содержать:*

*1. Функции формирования исходного файла;*

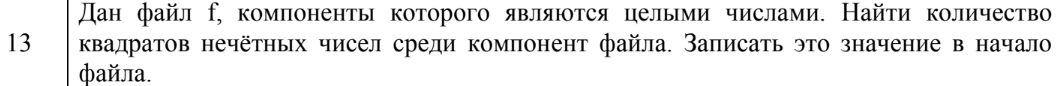
*2. Функции вывода результата работы программы;*

*3. Одну или более функций, реализующих вычислительную часть алгоритма.*

*Основная программа должна содержать только операторы вызова функций.*

*Вспомогательные файлы использовать нельзя! В задании предполагается, что нумерация*

*элементов файла начинается с 1.*

**

**РЕШЕНИЕ**

#include <stdio.h>

#include <math.h>

#include <string.h>

void fill\_file() { // функция для записи строки в исходный файл f

FILE\* fp = fopen("f.bin", "wb"); // открываем файл для бинарной записи

printf(" Input string of integer numbers separated by space:\n");

char str[256];

gets(str);

fputs(str, fp); // записываем строку в файл

fclose(fp);

}

int contain(char ch) { // функция проверяет, является ли символ допустимым для использования (цифры)

char aval[] = "1234567890"; // массив допустимых символов

int aval\_length = strlen(aval); // подсчитываем длину строки aval c помощью функции из библиотеки string.h

for (int i = 0; i < aval\_length; i++) {

if (aval[i] == ch) // если находим совпадение, то возвращаем 1 (правда - символ является допустимым)

return 1;

}

return 0; // если не будет возвращена 1, то возвращаем 0 (ложь)

}

void change\_file() { // функция для перезаписи файла

place:;

FILE\* fp = fopen("f.bin", "rb"); // поток для чтения из исходного бинарного файла

int count = 0; // переменная, в которой будет храниться число квадратов нечётных чисел

int num = 0; // текущее число

char ch = getc(fp); // переменная для хранения текущего символа из файла

char zero\_ascii = '0';

char str[256]; // строка, в которую считываем содержимое файла , но с ликвидацией повторяющихся разделительных пробелов

int str\_len = 0; // длина строки

int f1 = 0, f2 = 0; // флаги для работы с отрицательными числами

int f3 = 0; // флаг для отслеживания количества пробелов

while (ch != EOF) {

if (ch == ' ') { // различные условия для избавления записи в массив нескольких пробелов подряд

f3++;

if (f3 <= 1) {

str[str\_len] = ch;

str\_len++;

}

}

else {

f3 = 0;

str[str\_len] = ch;

str\_len++;

}

if (ch == '-' && num == 0) { // встречаем знак минус после пробела, то минус может подразумевать отрицательность

f1 = 1;

}

else if (ch != ' ') {

if (contain(ch)) { // если вдруг пользователь введёт недопустимый символ (не число), то просим его заново ввести строку

num = num \* 10 + (int)ch - (int)zero\_ascii; // преобразование символов в числа на разнице кодов чисел в таблиц ascii

if (f2) {

num \*= -1;

f2 = 0;

}

}

else {

printf("\nInput string again, couse you input not an integer number somewhere:\n");

fill\_file();

goto place; // начинаем функцию заново, так как у нас появилась новая строка и прошлые результаты нам не нужны

}

}

else {

if (sqrt(num) == (int)sqrt(num) && (int)sqrt(num) % 2 != 0) // проверяем, что число - квадрат нечётного числа

count++;

num = 0;

}

ch = getc(fp);

if (f1 && !contain(ch)) { // проверяем, если минус стоит не перед числом, то есть это недопустимый элемент

printf("\nInput string again, couse you input not an integer number somewhere:\n");

fill\_file();

goto place; // начинаем функцию заново, так как у нас появилась новая строка и прошлые результаты нам не нужны

}

else if (f1) {

f2 = 1; // ставим флаг, чтобы сделать число num отрицательным, чтобы оно не зачлось в качестве квадрата числа

}

f1 = 0;

}

if (sqrt(num) == (int)sqrt(num) && (int)sqrt(num) % 2 != 0) // проверка последнего числа

count++;

fclose(fp);

//теперь перезаписываем файл со значение переменной count в начале // через чистое позиционирование задача не решается

fp = fopen("f.bin", "wb");

fprintf(fp, "%d -> ", count); // записываем количество нечетных квадратов на первое место в файле с разделительной стрелочкой для красоты

for (int j = 0; j < str\_len; j++)

putc(str[j], fp);

fclose(fp);

}

void print\_file() { // функция для вывода содержимого файла

FILE\* fp = fopen("f.bin", "rb");

char ch;

printf("\n Answer:\n");

while ((ch = getc(fp)) != EOF)

printf("%c", ch);

fclose(fp);

}

int main(void) {

// используем наши функции в нужном порядке для решения задачи

fill\_file();

change\_file();

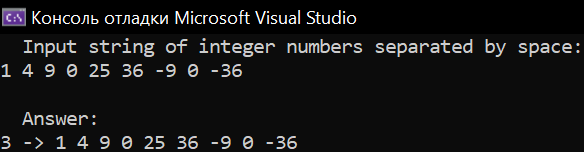
print\_file();

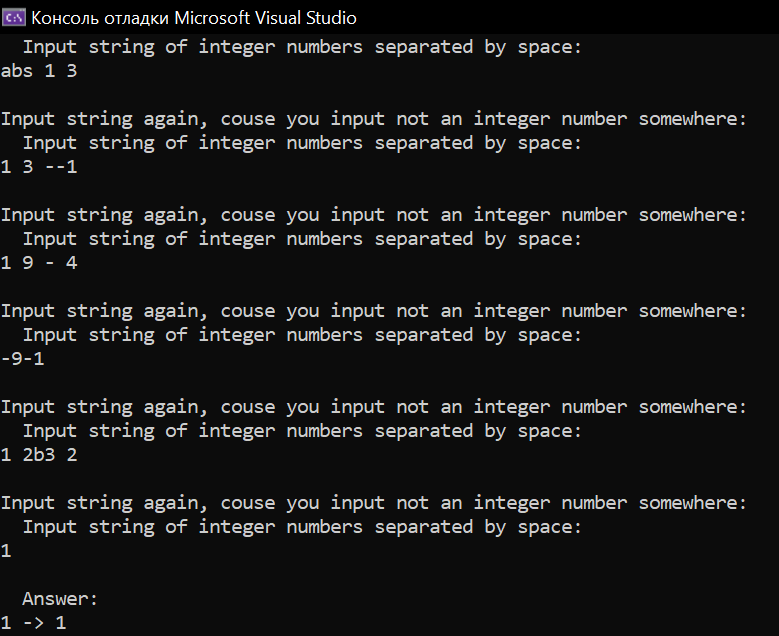
return 0;

}

**ТЕСТЫ**

**Тест № 1** // обычный пример

****

**Тест № 2** // пример содержит все возможные обрабатываемые исключения

**Тест № 3** // избавляемся от лишних пробелов

**Изображение выглядит как текст

Автоматически созданное описание**