Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Инженерно-психологическое

обеспечение информационных технологий»

Учебная дисциплина «Основы алгоритмизации и программирования»

Отчет

по лабораторной работе No8

«Функции»

Вариант 25

Подготовил:

Подрябинкин З.Д. гр. 410902

Проверил: Усенко Ф.В.

Минск 2024

Цель: сформировать навыки и умения обработки структурированных типов данных, организованных в виде функций.

Задание1: (Вариант **25**)Напишите программу деления двух обыкновенных несократимых дробей  и  Результат представить в виде несократимой дроби.

Задание2: (Вариант **25**)

а) для последовательности удаляет все четные элементы;

б) для последовательности удаляет все элементы, заключенные между двумя нулевыми элементами.

Далее представлены коды, схемы и результаты работ

Программ 1 (задание 1) и 2 (задание 2)

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

struct drob

{

int chisl;

int znam;

};

drob chastnoye(drob inp1, drob inp2) //функция вычисления частного

{

drob outp;

int znak = 1; // знак дроби

outp.chisl = inp1.chisl \* inp2.znam;

outp.znam = inp1.znam \* inp2.chisl;

if (((outp.chisl > 0) and (outp.znam < 0)) or ((outp.chisl < 0) and (outp.znam > 0))) // определение итогового знака дроби

{

znak = -1;

}

outp.chisl = pow(pow(outp.chisl, 2), 0.5); // взятие модуля от дроби для правильной работы алгоритма

outp.znam = pow(pow(outp.znam, 2), 0.5);

for (int i = 2; (i <= outp.chisl) and (i <= outp.znam); i++) //привидение дроби к стандартному виду

{

if ((outp.chisl % i == 0) and (outp.znam % i == 0))

{

outp.chisl = outp.chisl / i;

outp.znam = outp.znam / i;

i = 2;

}

}

outp.chisl \*= znak; // возврат итогового знака дроби

if (outp.znam == 0)

{

cout << "\n !!!! \n В знаменателе обнаружен 0. Вычисления не будут являться правильными. Перезапустите програму и введите данные заново,";

cout << "избегая нуля в знаменателях и числителе второй дроби. \n !!!!\n";

}

return outp;

}

drob vvod() //функция ввода дробного числа

{

drob res;

int chisl = 0;

int znam = 0;

char input[256];

int znak = 1;

int funct\_status = -1; // переменная для проверки вида дроби и поиска места символа / в записи дроби

cin.getline(input, 256);

for (int i = 0; i < strlen(input); i++) // поиск знака / в дроби

{

if (input [i] == '/')

{

funct\_status = i;

}

}

if (input[0] == '-') // проверка на отрицательное число в числителе

{

znak \*= -1;

}

if (input[funct\_status + 1] == '-' and funct\_status != -1) // проверка на отрицательное число в знаменателе

{

znak \*= -1;

}

if (funct\_status == -1)

{

znam = 1;

for (int i = 0; i < strlen(input); i++) // действие на случай если отсутствует знак /

{

if (input[i] != '-') //проверка на знак минус и перевод числа из массива char в int

{

chisl += ((int)input[i] - '0') \* pow(10, strlen(input) - i - 1);

}

}

cout << endl << "Отсутствует символ \"/\". Будет выведена дробь " << znak \* chisl << "/1" << endl;

}

else

{

for (int i = 0; i < funct\_status; i++)

{

if (input[i] != '-') //проверка на знак минус и перевод числа из массива char в int

{

chisl += ((int)input[i] - '0') \* pow(10, funct\_status - i - 1);

}

}

for (int i = funct\_status +1 ; i < strlen(input) ; i++)

{

if (input[i] != '-') //проверка на знак минус и перевод числа из массива char в int

{

znam += ((int)input[i] - '0') \* pow(10, strlen(input) - i - 1);

}

}

}

res.chisl = chisl \* znak;

res.znam = znam;

if (znam == 0)

{

cout << "\n !!!! \n В знаменателе обнаружен 0. Вычисления не будут являться правильными. \n !!!! \n";

}

return res;

}

void main()

{

setlocale(LC\_ALL, "ru");

drob a;

drob b;

cout << endl << "Введите первую дробь через \"/\" (например 53/21): ";

a = vvod();

cout << endl << "Введите вторую дробь через \"/\" (например -25/2, избегайте нулевой дроби): ";

b = vvod();

drob res;

res = chastnoye(a, b);

cout << "\nРезультат деления: (" << a.chisl << '/' << a.znam << ") / (" << b.chisl << '/' << b.znam << ") = ";

cout << res.chisl << '/' << res.znam << endl;

system("pause");

}

Код программы 1

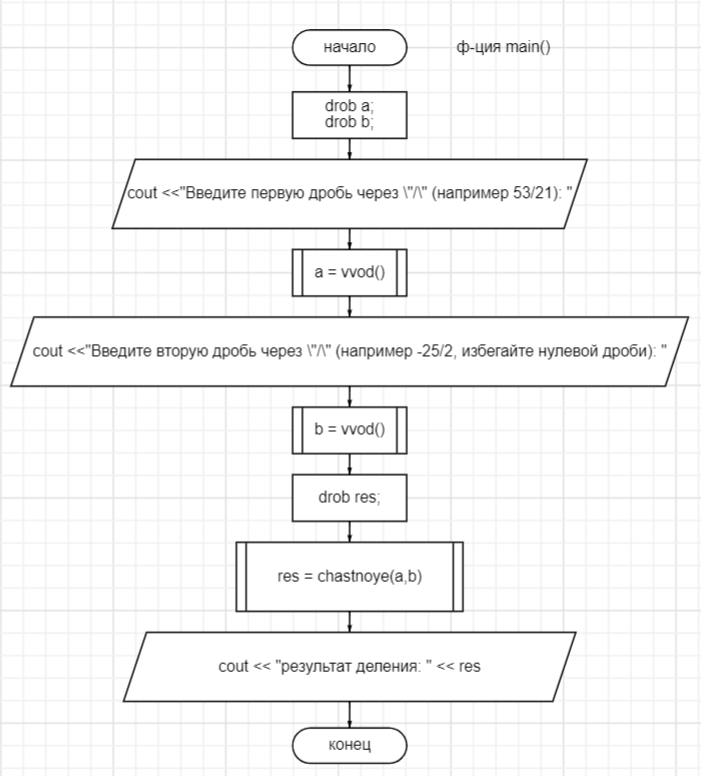


Схема кода программы 1

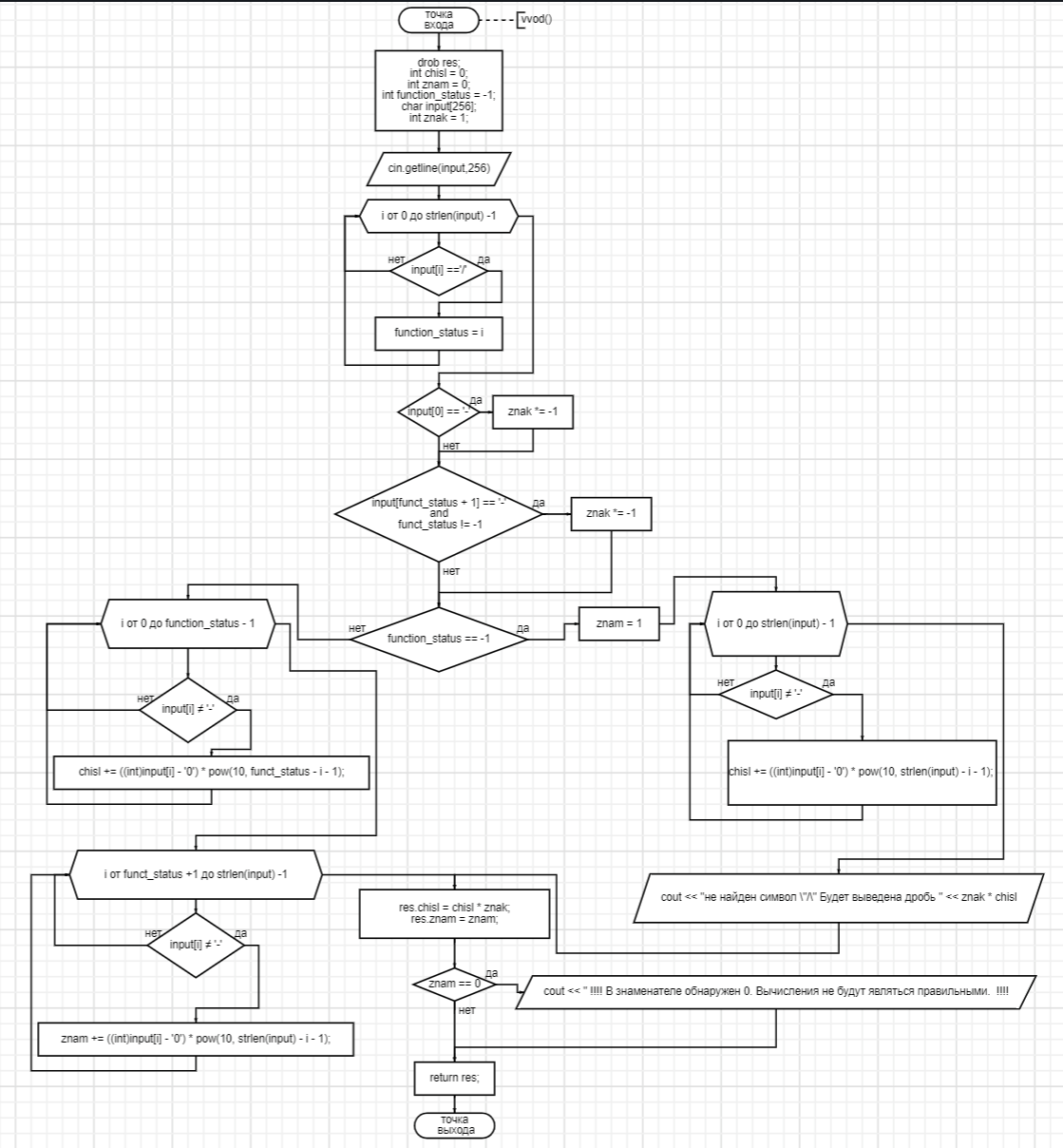


Схема функции ввода

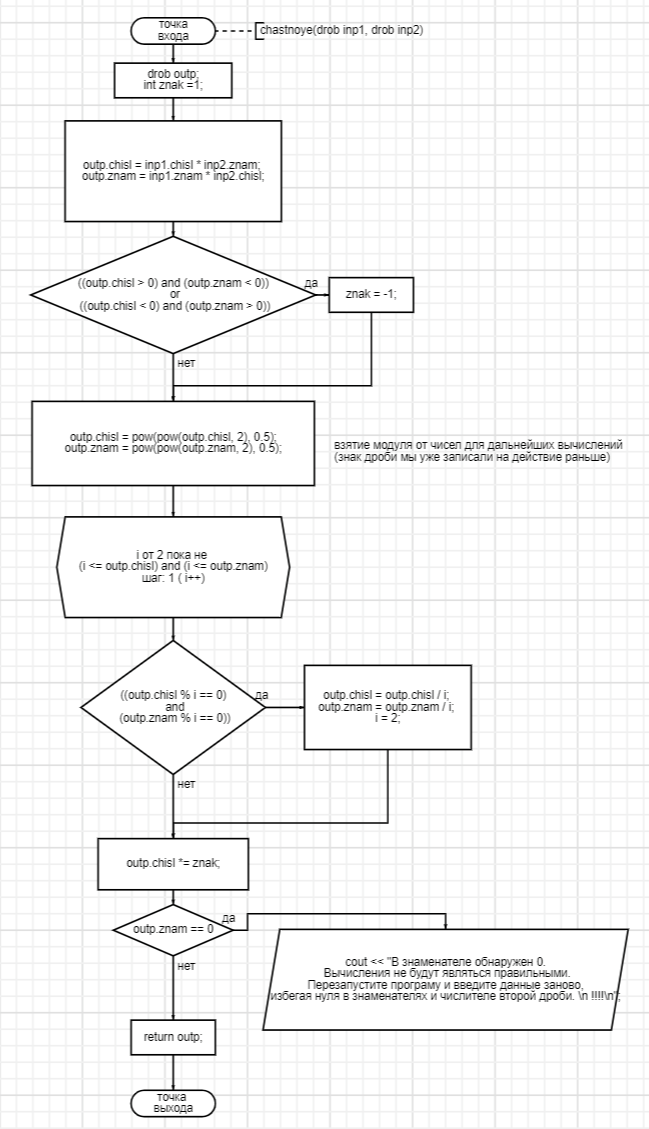


Схема функции вычисления частного

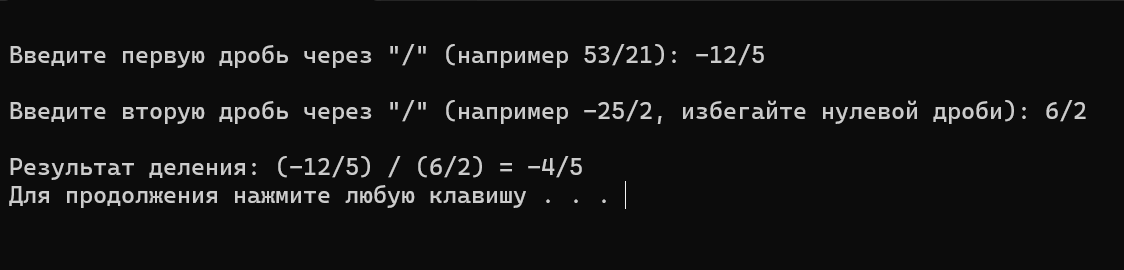


Рисунок 1 – Скриншот результата работы программы1.

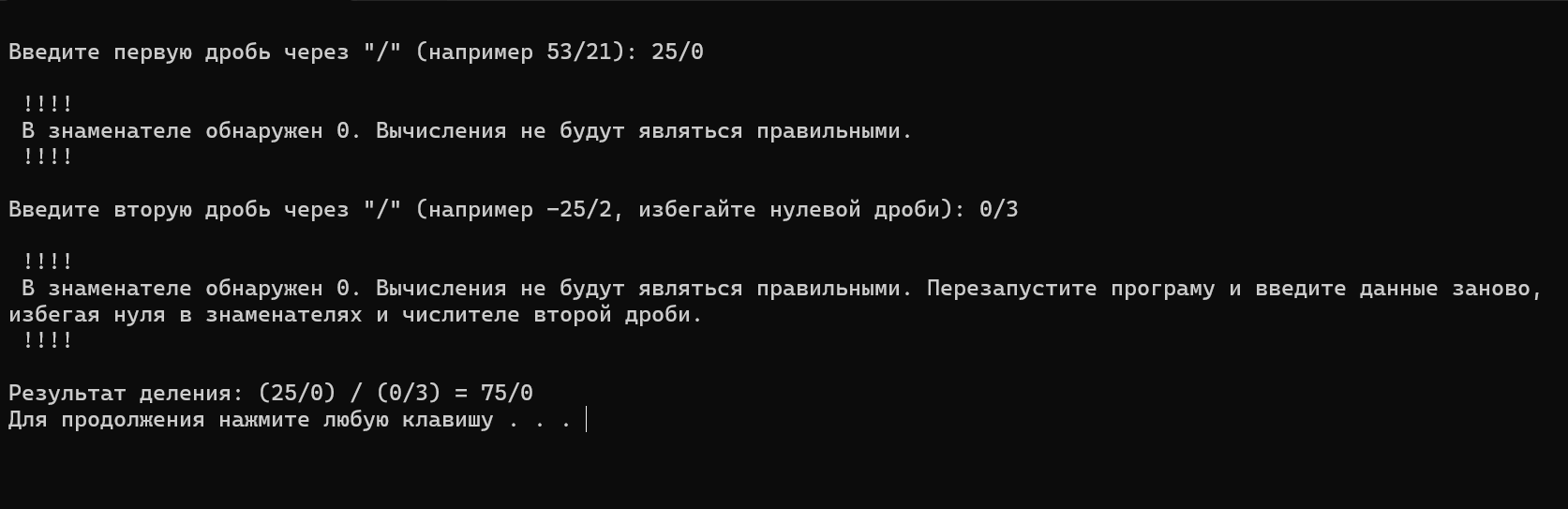


Рисунок 2 – Скриншот результата работы программы1 в случае ввода некорректной дроби.

#include<iostream>

#include<string>

using namespace std;

void remove(char input[])

{

for (int i = 0; input[i] != '\0'; i++)

{

if (input[i] == '\n')

{

for (int j = i; input[j] != '\0'; j++)

{

input[j] = input[j + 1];

}

i--;

}

}

}

void preobr(char input[], int inp\_length)

{

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

{

if ((i - 1) % 2 == 0)

{

input[i] = '\n';

}

}

remove(input);

}

void preobr(char input[], int inp\_length, char nnul)

{

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

{

if (input[i] == nnul)

{

input[i] = '\n';

i++;

for (i; input [i] != nnul and i < inp\_length; i++)

{

input[i] = '\n';

}

if (input[i] == nnul)

{

input[i] = '\n';

}

}

}

remove(input);

}

void main()

{

setlocale(LC\_ALL, "ru");

char input[256];

cout << "Введите нейкую строку символов на латинском: ";

cin.getline(input, 256);

cout << "Введите код операции: \n 1 -- убрать все чётные символы \n 2 -- убрать все элементы между нулями \n другой символ -- закрыть программу. \n";

char answ;

cin >> answ;

switch (answ)

{

case '1':

preobr(input, strlen(input));

break;

case '2':

preobr(input, strlen(input), '0');

break;

default:

cout << "выход";

break;

}

cout << endl << "Вывод текста: ";

cout << input;

cout << endl;

system("pause");

}

Код программы 2

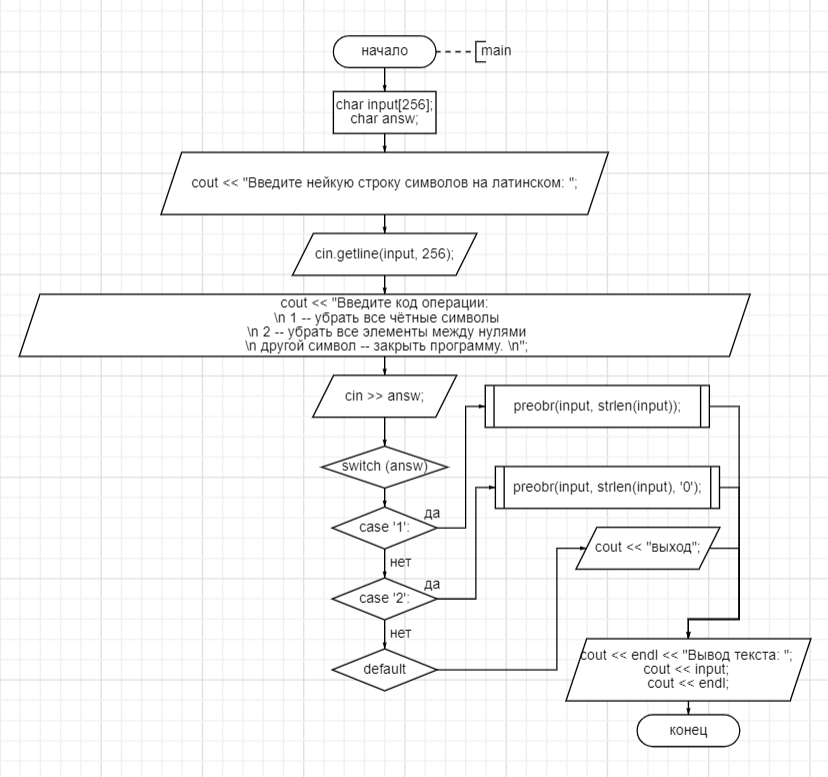


Схема кода программы 2

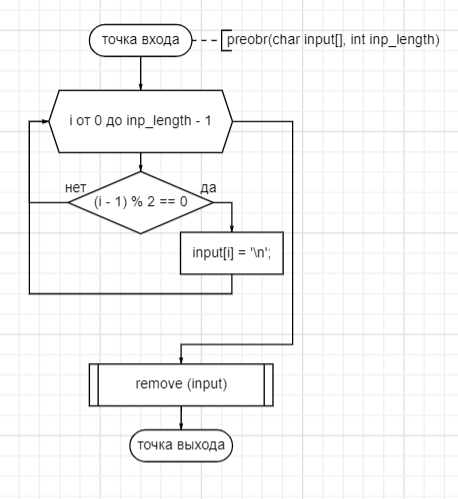


Схема функции первого преобразования программы2

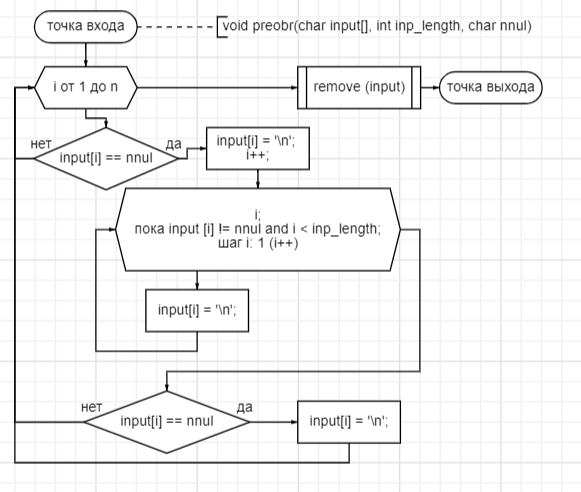


Схема Скриншот кода функции второго преобразования программы2

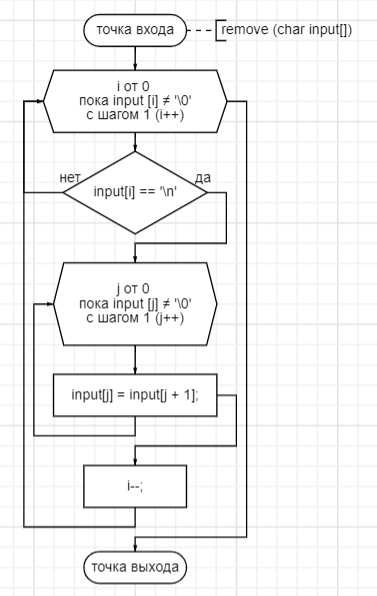


Схема функции удаления текста программы2

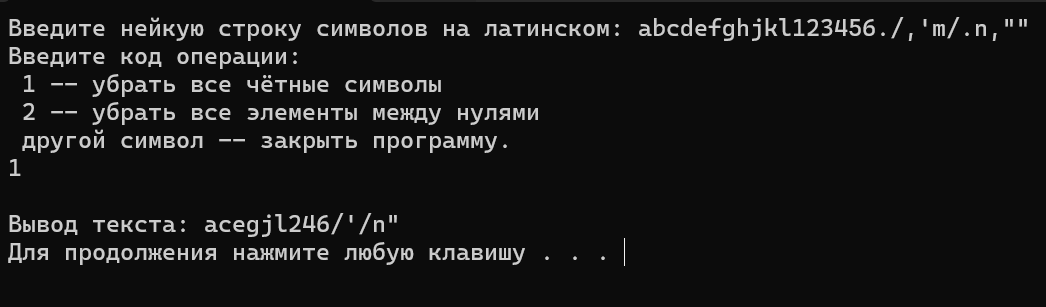


Рисунок 3 – Скриншот результата работы программы2 (пользователем выбрана первая функция преобразования).

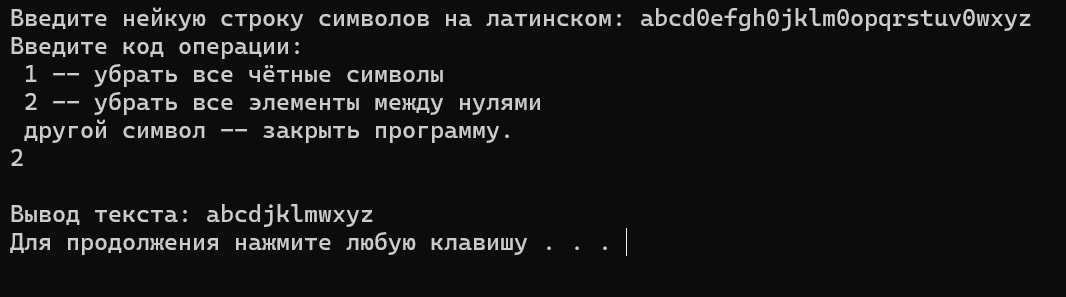


Рисунок 4 – Скриншот результата работы программы2 (пользователем выбрана вторая функция преобразования).

Вывод: В результате работы были созданы две программы содержащие функции. Причём в первой работе для возврата сразу нескольких значений от функции были использованы структуры. Помимо этого в первой программе реализована базовая проверка на неправильный ввод дроби. Во второй программе были созданы 2 перегруженные функции выполняющие условия данные в задании и одна функция для быстрого удаления лишних элементов строки со смещением. Во второй программе в качестве аргумента для функции наряду с обычными значениями используется указатель на массив нуль-терминированной строки. Поскольку все преобразования такой строки даже внутри функции носят глобальный характер, все функции могут иметь тип void и не возвращают какие-либо значения. Использование функций позволило облегчить читаемость программы, а так же уменьшить количество строк кода.