Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

**ОТЧЕТ**

по учебной практике

на кафедре Прикладной Математики и Кибернетики

Вариант 4

Выполнил:

студент гр. ИП-713 \_\_\_\_\_\_\_\_\_\_\_\_\_/Михеев Н.А/

«20» июня 2019 г.

Руководитель практики

доцент каф. ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_/Разинкина Т.Э./

«20» июня 2019 г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2019 г.

Оглавление

1. Постановка задачи3
2. Описание алгоритма3
3. Программная реализация4
4. Результат работы6

Используемые источники11

# Постановка задачи

**Текст 4 Генерация подмножеств**

Задано целое положительное число n, которое представляет собой мощность некоторого множества. Требуется с минимальными трудозатратами генерировать все подмножества этого множества, для чего каждое последующее подмножество должно получаться из предыдущего путем добавления или удаления только одного элемента. Множество и все его подмножества представляются битовой шкалой. Для генерации использовать алгоритм построения бинарного кода Грея.

В качестве результата выводить построчно каждое из подмножеств (в виде битовой шкалы), сопровождая их порядковыми номерами.

Оценка «хорошо»:

В дополнение к предыдущему, в случае большого количества результирующих строк (превышающего размер экрана) выполнять поэкранную выдачу, а также осуществлять их вывод в файл с выдачей на экран сообщения для пользователя – имя файла, его местонахождение…

1. Описание алгоритма

**void p\_count(int n) –** функция подсчета p = Q(i) – число, на единицу превышающее количество «2» в разложении числа i на множители

**void dec2bin(int x, int n)** – функция перевода кода грея в его бинарное представление

**void gray\_code(int n) –** основная функция, занимающаяся получением i-го кода грея, получающая на вход заданное пользователем число n – мощность множества

Пользователь вводит число n >= 0, являющееся мощностью желаемого множества. При некорректных значениях пользователю даётся возможность переввести значения. После генерации всех множеств на экране появляется сообщение с вопросом: «необходимо ли вывести полученные множества?» Пользователь может ответить Y или N (yes или no). Если пользователь решил посмотреть на полученный результат, то программа печатает полученные множества, но не больше 20 за раз, можно ввести любой символ чтобы продвинуться на следующие 20 множеств или ввести «0» чтобы прервать вывод. Далее пользователь возвращается в начальное меню, откуда он может выйти из программы или сгенерировать новые подмножества.

1. Программная реализация

#include <bits/stdc++.h>

using namespace std;

ofstream filein;

ifstream fileout;

void p\_count(int n)

{

int div = 2, dvoiki = 0;

while (n > 1)

{

while (n % div == 0)

{

if(div == 2)

dvoiki++;

n = n / div;

}

if (div == 2)

{

div++;

}

else div += 2;

}

filein << dvoiki + 1 << "\t-\t";

}

void dec2bin(int x, int n)

{

int\* binaryNumber = new int(x);

int i = 0;

while(x > 0)

{

binaryNumber[i] = x % 2;

x = x / 2;

i++;

}

for(int j = 0; j < n - i; j++)

{

filein << "0 ";

}

for(int j = i - 1; j >= 0; j--)

{

filein << binaryNumber[j] << " ";

}

filein << endl;

delete binaryNumber;

}

void gray\_code(int n)

{

char button;

int k = 0, i = 0;

filein.open("Gray\_Codes.txt", ofstream::out);

filein << "Gray codes for n = " << n << " bit" << endl;

filein << "i\t \tp\t\tB" << endl;

int N = 1 << n;

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

{

filein << i << "\t-\t";

p\_count(i);

int x = i ^ (i >> 1);

dec2bin(x, n);

}

filein.close();

cout << "Gray codes for n = " << n << " were generated and saved into \"...\\Gray\_Codes.txt\"" << endl;

cout << "Would you like to see them?" << endl << "Print: <Y>/<N>" << endl;

fileout.open("Gray\_Codes.txt", ifstream::in);

cin >> button;

i = 0;

if(button == 'Y' || button == 'y')

{

do

{

system("cls");

k = i + 20;

for(i; i < k; i++)

{

string str;

if(i == N)

{

cout << "End of the table." << endl;

fileout.close();

return;

}

getline(fileout, str);

cout << str << endl;

}

cout << "<Any key> - next 20, <0> - exit." << endl;

cin.ignore(0);

cin >> button;

if(button == '0')

break;

}

while(i < N);

}

fileout.close();

}

int main()

{

int n;

while(1)

{

system("cls");

cout << "Enter your Gray\_code power or 0 to exit: ";

cin >> n;

if(n == 0)

break;

else if(n > 15 || n < 0)

{

cout << "Wrong number! Try again." << endl;

system("PAUSE");

}

else

{

gray\_code(n);

system("PAUSE");

}

}

return 0;

}

1. Результат работы

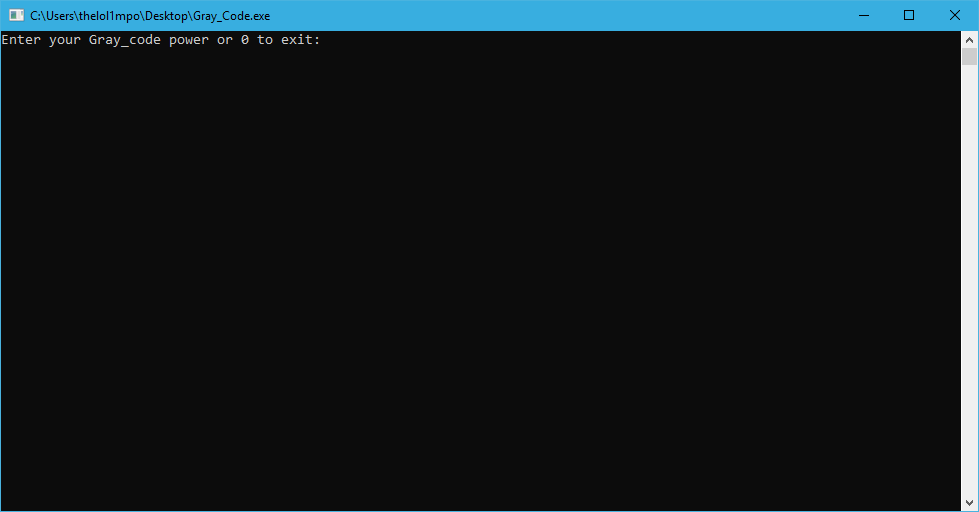


Рис.1 Начальное меню

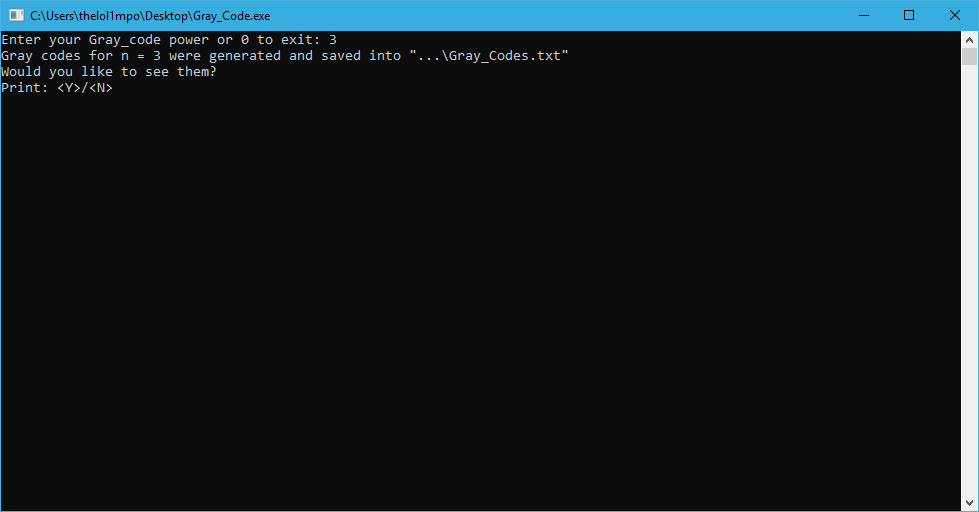


Рис.2 Ввод n = 3, сообщение о успешной генерации кодов и сохранении в файл Gray\_Codes.txt, вопрос о том, нужно ли вывести полученные подмножества?

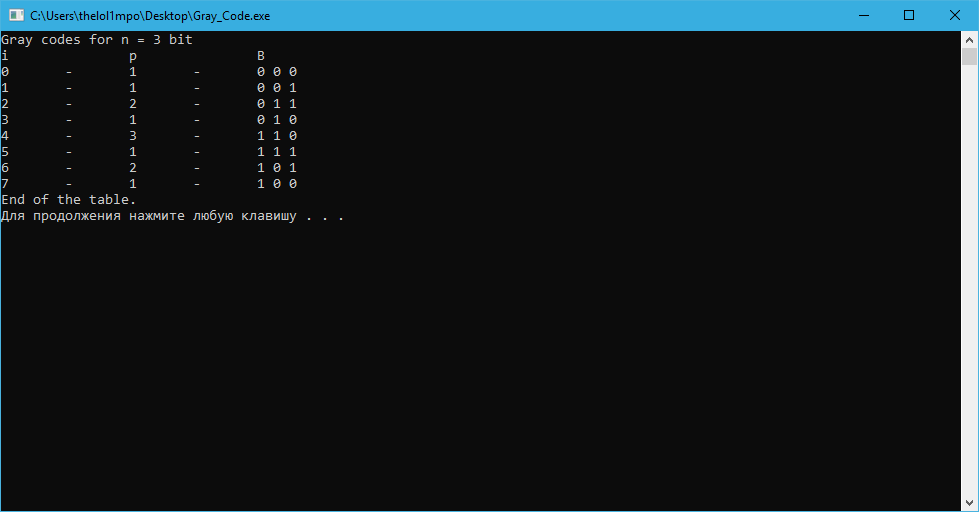


Рис.3 Вывод кодов Грея для n = 3

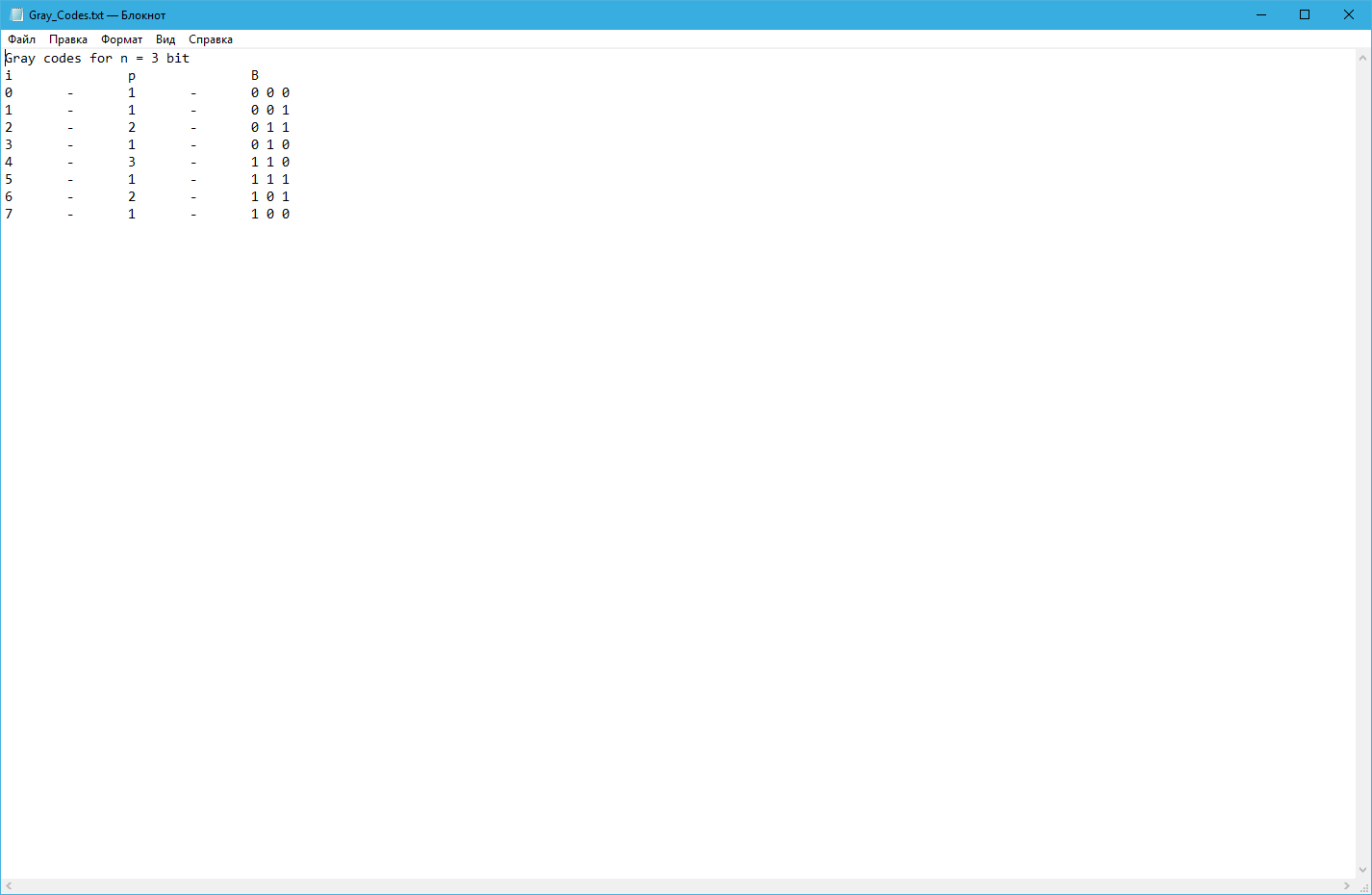


Рис.4 txt-файл с успешно записанными подмножествами

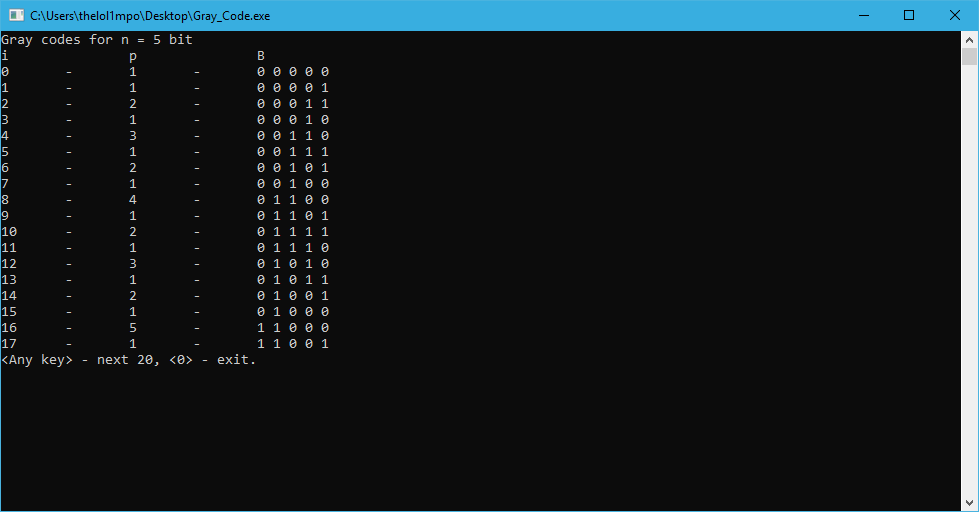


Рис.5 Тест работы программы №2

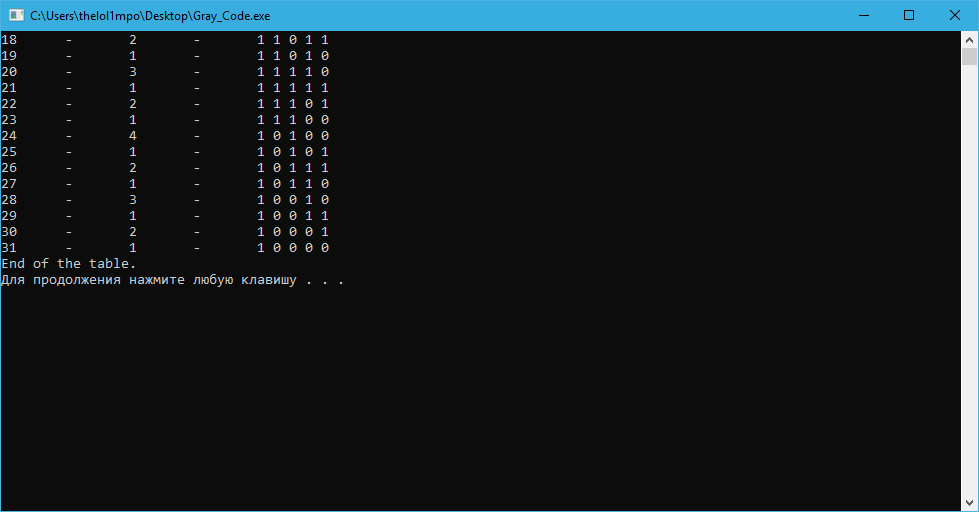


Рис.5 Последняя страница вывода подмножеств теста №2

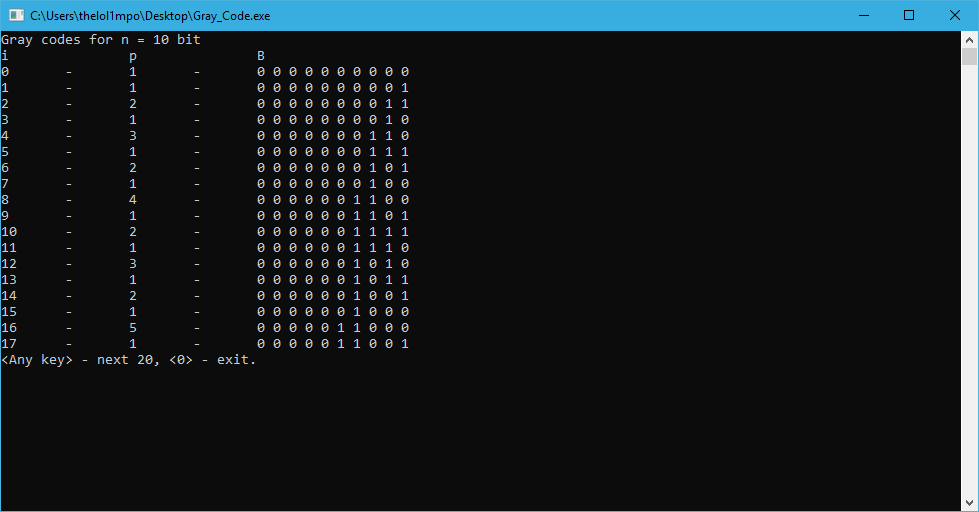


Рис.6 Тест работы программы №3

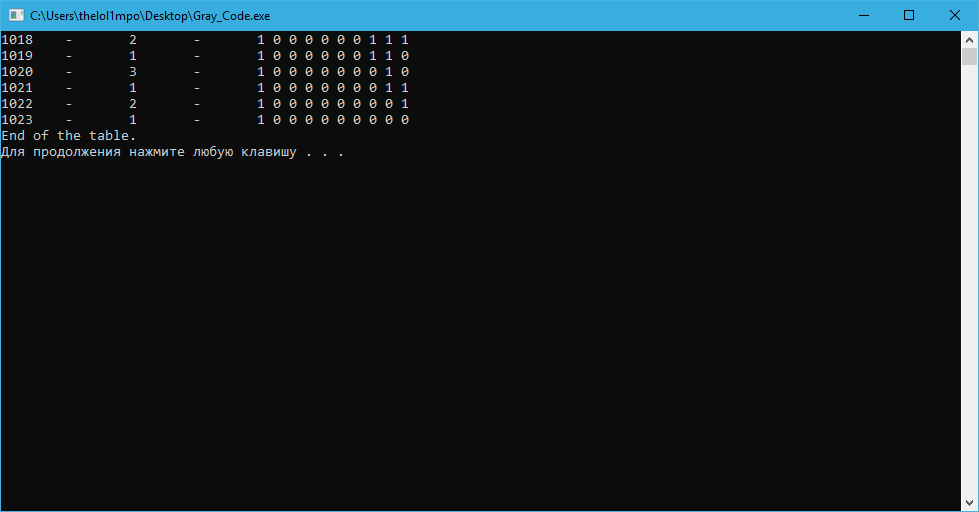


Рис.7 Последняя страница вывода подмножеств теста №3

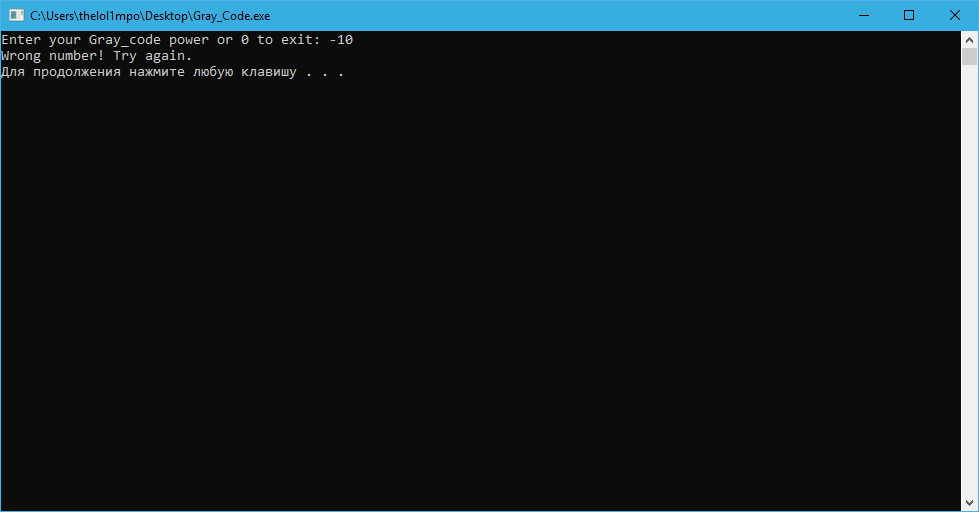


Рис.8 Тест на ввод отрицательного значения

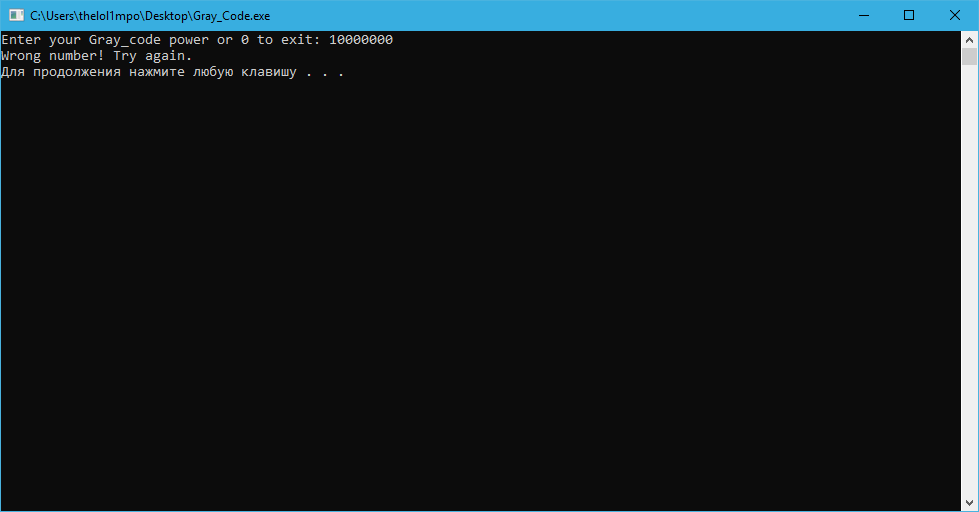


Рис.9 Тест на ввод слишком большого значения

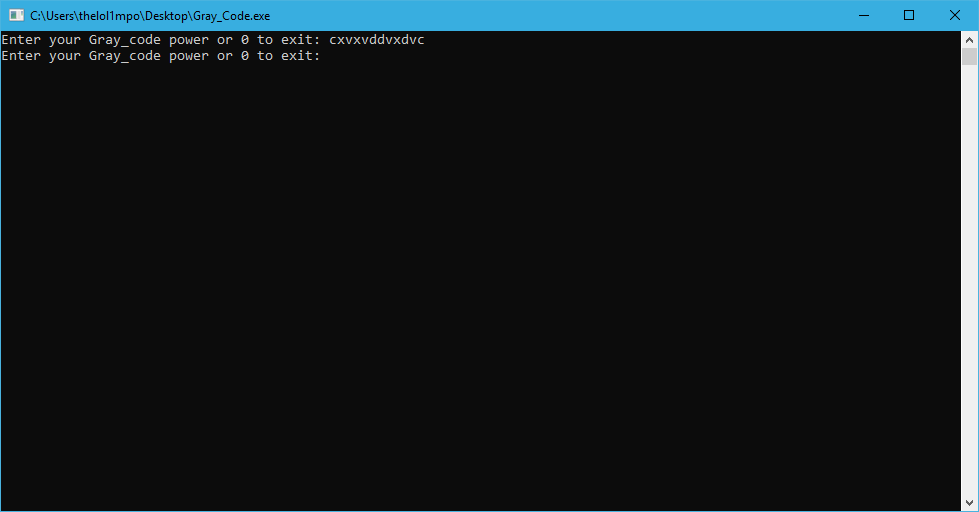


Рис.10 Тест на ввод некорректных данных

Используемые источники

1. <Http://e-maxx.ru/algo>
2. “C++ Базовый курс” - Г. Шилдт
3. “Объектно-ориентированное программирование в C++” – Р. Лафоре
4. <http://neerc.ifmo.ru/wiki/index.php>