|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **По домашней работе №** | 2 |

**Название:**

Программирование на С++

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  | И.А. Люляев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

***Часть 2.1. Функции***

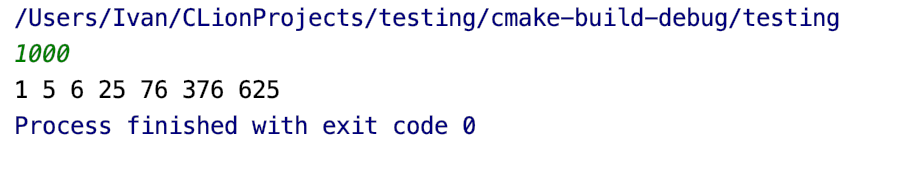
Автоморфными называют числа, которые содержатся в последних разрядах их квадратов. Например: 52 = 25, 252 = 625.

Написать программу, определяющую автоморфные числа в диапазоне 1 до n (n вводится с клавиатуры, n >100). Вывести на печать все найденные числа. При программировании использовать функции.

**Код программы:**

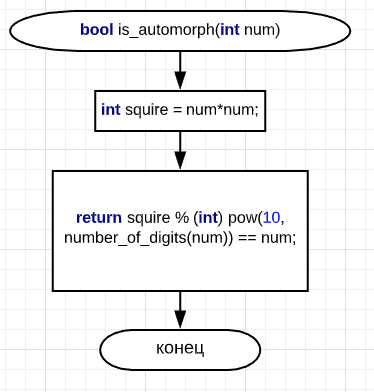
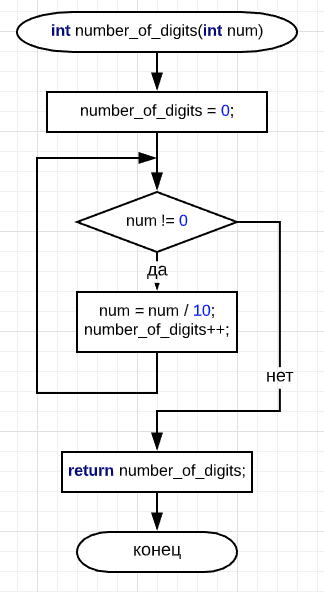
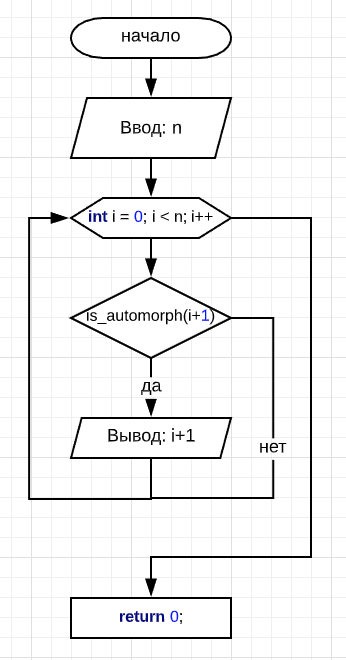
#include **<iostream>**#include **<cmath>  
  
using namespace** std;  
  
  
**int** number\_of\_digits(**int** num) {  
 **int** number\_of\_digits = 0;  
 **while** (num != 0) {  
 num = num / 10;  
 number\_of\_digits++;  
 }  
 **return** number\_of\_digits;  
}  
  
**bool** is\_automorph(**int** num) {  
 **int** squire = num\*num;  
 *// если последние цифры квадрата заданного числа совпадают с самим числом, то true* **return** squire % (**int**) pow(10, number\_of\_digits(num)) == num;  
}  
  
**int** main() {  
 **int** n;  
 cin >> n;  
 **for** (**int** i = 0; i < n; i++) {  
 **if** (is\_automorph(i+1)) cout << i+1 << **" "**;  
 }  
 **return** 0;  
}

**Результаты:**

****

**Вывод:** по результатам выполнения первой части домашнего задания я создал успешно работающую программу на С++ с использованием функций.

**Схема алгоритма:**



***Часть 2.2. Текстовая обработка***

Дана последовательность строк. Строки содержат слова, разделенные пробелом. Используя цифровой шифр, например 31206, зашифровать каждую строку по следующей методике:

Viral lonely to sea

31206 312061 12 063

Yjtar opperz uq skd

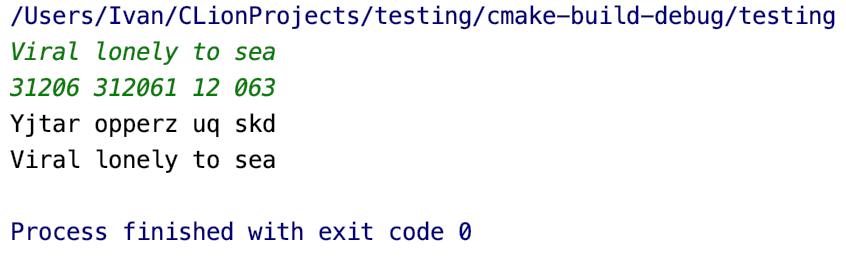
то есть к каждой букве применяют соответствующую цифру для определения смещения этой буквы по алфавиту в шифре (V+3=Y, O+1=P и т.д.).

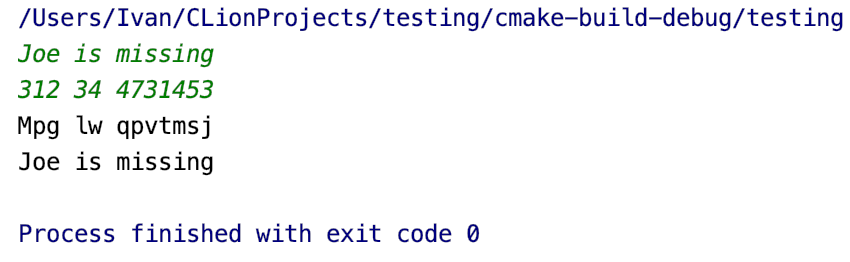
Написать программу, обеспечивающую ввод строк, шифровку и дешифровку. Вывести на печать зашифрованную и подвергнутую дешифровке последовательности строк.

**Код программы:**

#include **<iostream>**#include **<stdlib.h>**#include **<string>  
  
typedef struct** coding {  
 std::string st;  
 std::string code;  
 std::string encrypted\_st;  
}coding;  
  
**void** st\_input(coding &code) {  
 getline(std::cin, code.st);  
 getline(std::cin, code.code);  
}  
  
std::string encryption(coding &code) {  
 **for** (**int** i = 0; i < code.st.size(); i++) {  
 **char** t = code.code[i];  
 **if** ((code.st[i] + atoi(&t)) < 127) {  
 code.encrypted\_st.push\_back((**char**)((**int**)code.st[i] + (**int**)atoi(&t)));  
 }  
 }  
 std::cout << code.encrypted\_st << std::endl;  
 **return** code.encrypted\_st;  
}  
  
std::string decryption(coding &code) {  
 std::string decrypted;  
 **for** (**int** i = 0; i < code.encrypted\_st.size(); i++) {  
 **char** t = code.code[i];  
 **if** (code.encrypted\_st[i] != 95) decrypted.push\_back((**char**)((**int**)code.encrypted\_st[i] - (**int**)atoi(&t)));  
 }  
 std::cout << decrypted << std::endl;  
 **return** decrypted;  
}  
  
  
**int** main() {  
 coding code = {**""**, **""**, **""**};  
 st\_input(code);  
  
 encryption(code);  
  
 decryption(code);  
  
 **return** 0;  
}

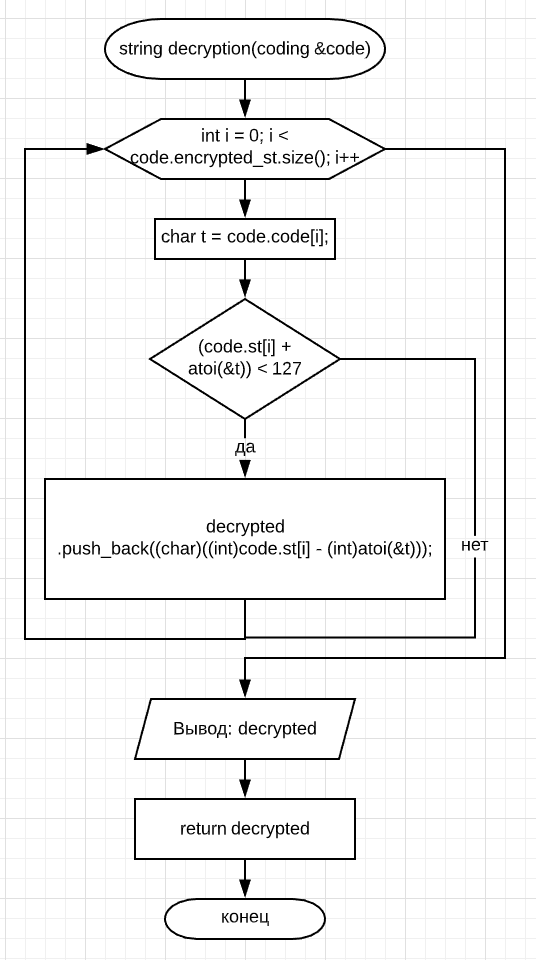
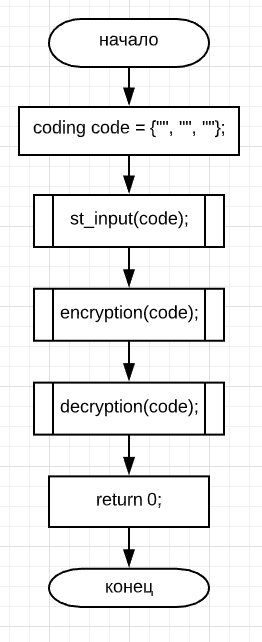
**Результаты:**

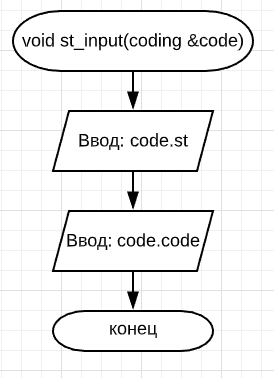
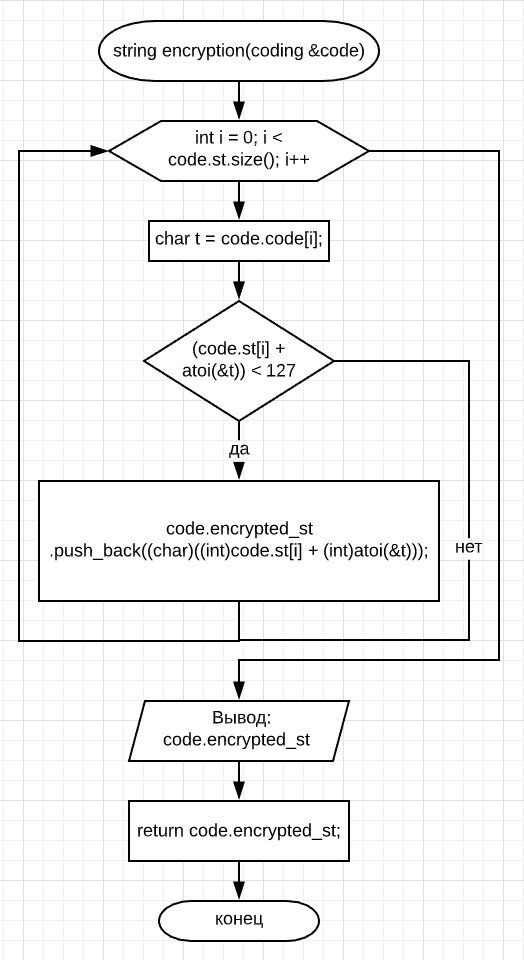
****

****

**Вывод:** по результатам выполнения второй части домашнего задания я продуктивно попрактиковался в текстовой обработке в специфике языка С++

**Схема алгоритма:**

****

****

***Часть 2.3. Файлы***

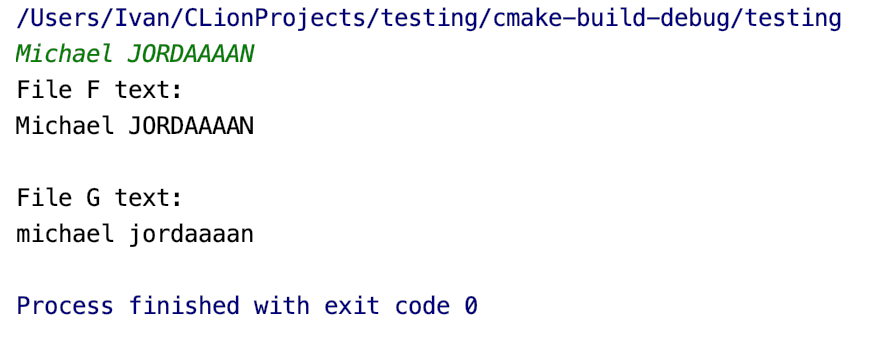
Организовать программным способом символьный файл ***F***. Получить файл ***G***,образованныйизфайла ***F*** заменойвсехегопрописных(больших) букв одноименными строчными(малыми).

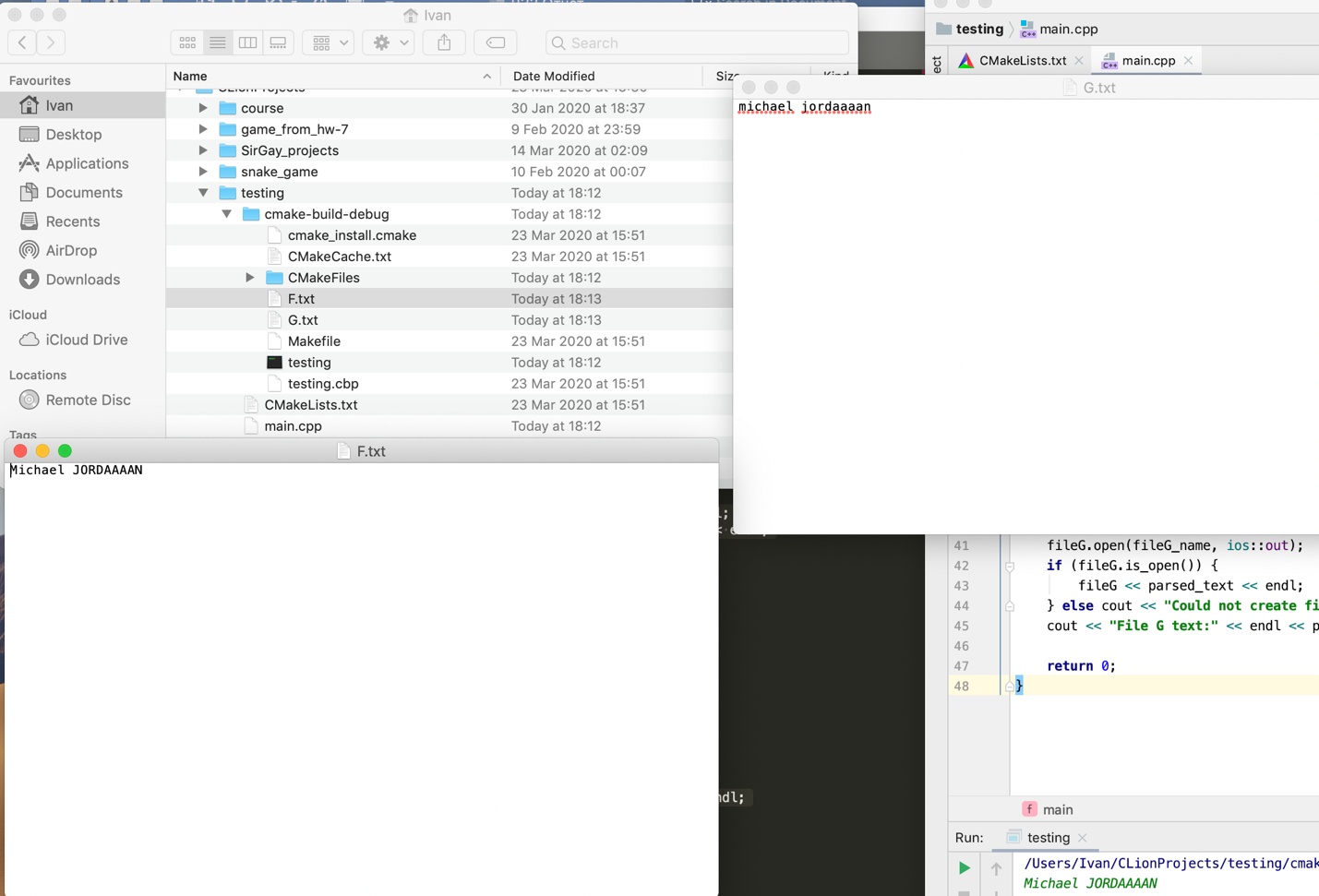
Вывести на экран оба файла.

**Код программы:**

#include **<iostream>**#include **<fstream>  
  
using namespace** std;  
  
**int** main() {  
 *// reading line to write to initial file* string text;  
 getline(cin, text);  
  
 *// creating initial file F* fstream fileF;  
 string fileF\_name = **"F.txt"**;  
 fileF.open(fileF\_name, ios::out);  
 **if** (fileF.is\_open()) {  
 fileF << text << endl;  
 fileF.close();  
 } **else** cout << **"Could not create file: "** << fileF\_name << endl;  
  
 *// reading from file F* string text\_from\_fileF;  
 fileF.open(fileF\_name, ios::in);  
 **if** (fileF.is\_open()) {  
 getline(fileF, text\_from\_fileF);  
 fileF.close();  
 } **else** cout << **"Could not open file: "** << fileF\_name << endl;  
 cout << **"File F text:"** << endl << text\_from\_fileF << endl << endl;  
  
 *// parsing text from F to write it in G* string parsed\_text = text\_from\_fileF;  
 **for** (**char** & i : parsed\_text) {  
 **if** (((**int**)i >= 65) && ((**int**)i <= 90)) {  
 i = (**char**)((**int**)i + 32);  
 }  
 }  
  
 *// writing parsed text to G and printing that text* fstream fileG;  
 string fileG\_name = **"G.txt"**;  
 fileG.open(fileG\_name, ios::out);  
 **if** (fileG.is\_open()) {  
 fileG << parsed\_text << endl;  
 } **else** cout << **"Could not create file: "** << fileG\_name << endl;  
 cout << **"File G text:"** << endl << parsed\_text << endl;  
  
 **return** 0;  
}

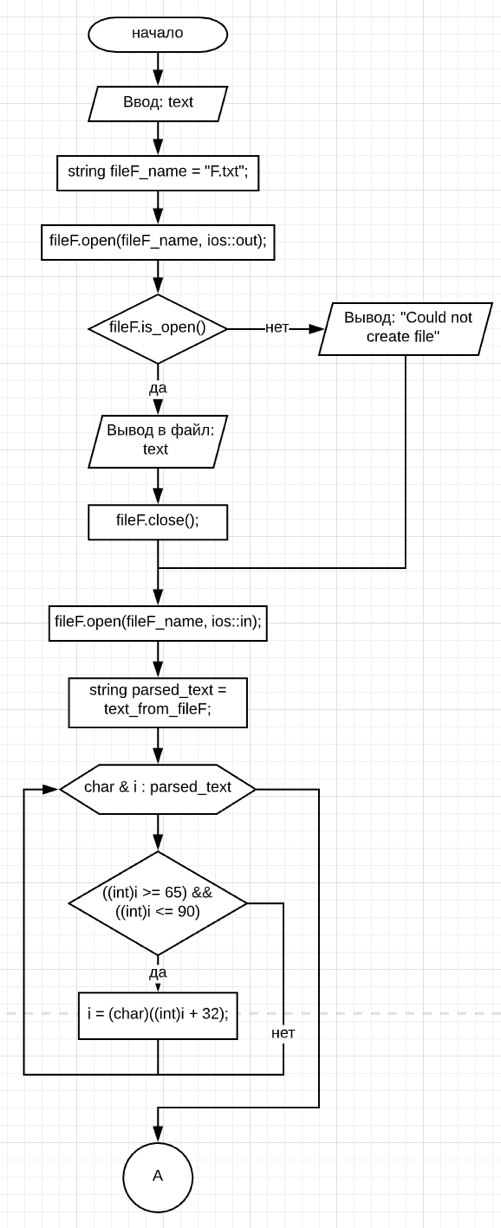
**Результаты:**

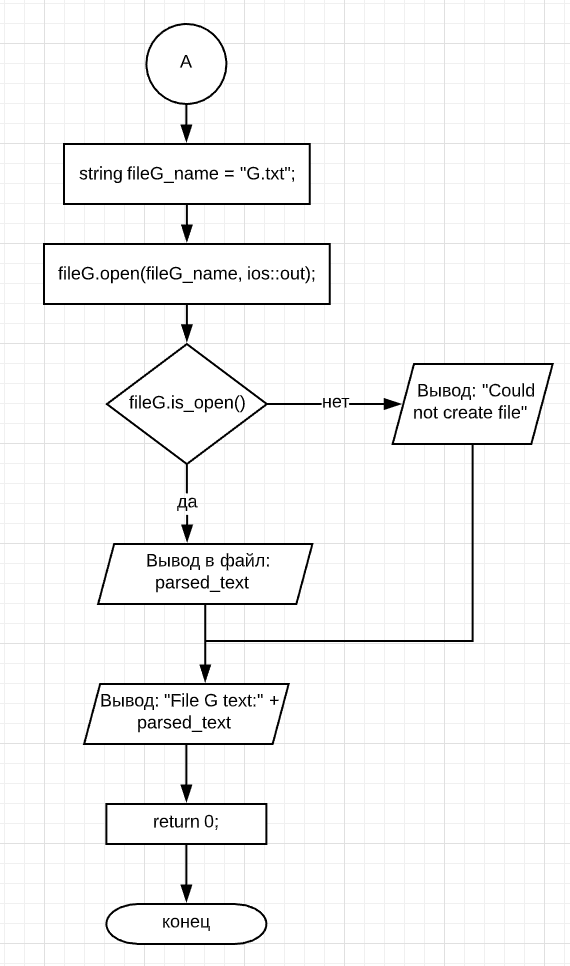
****

****

**Вывод:** по результатам выполнения третьей части домашнего задания я освоил азы работы с файлами в специфике Языка С++

**Схема алгоритма:**

****

****