

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей  
Кафедра программного обеспечения информационных технологий  
Дисциплина: Теория информации (ТИ)

**ОТЧЕТ**  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Тема работы:  
КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

Выполнила  
студентка: гр. 351002

Хмель А.А.

Проверила:

Болтак С.В.

Минск 2025

## СОДЕРЖАНИЕ

1 Задание .....	3
2 Тестовые наборы .....	4
3 Пример работы алгоритма.....	15

## 1 ЗАДАНИЕ

### Вариант 2

Реализовать шифратор и дешифратор *алгоритма Эль-Гамала* файла с произвольным содержимым, используя алгоритм быстрого возведения в степень, а также реализовать вычисление открытого ключа  $g$  при данном значении  $p$ , используя алгоритм нахождения первообразного корня по модулю. Значения параметров  $p$ ,  $x$  и  $k$  задаются пользователем. Программа должна осуществлять проверку ограничений на вводимые пользователем значения параметров алгоритма. Организовать вывод содержимого зашифрованного файла на экран в виде чисел в 10-й системе счисления. Вывести значение  $g$  на экран. Результат работы программы – зашифрованный/расшифрованный файл/ы.

Используя алгоритм из методички, искать все первообразные корни по модулю  $p$ . Все найденные корни вывести на экран и предложить для шифрования ввести на выбор любой из найденных.

При использовании длинной арифметики для определения простоты числа использовать один из вероятностных тестов: тест Ферма или тест Миллера-Рабина.

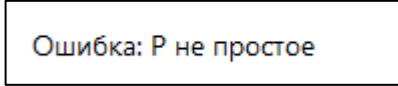
## 2 ТЕСТОВЫЕ НАБОРЫ

### Тест 1

Тестовая ситуация: проверка на корректность вводимого числа  $p$ .

Исходные данные:  $p = 35$  (не простое число).

Полученный результат: ошибка о том, что число не простое.



Ошибка: P не простое

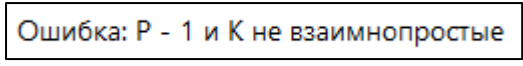
Рисунок 1 – Полученный результат. Тест 1

### Тест 2

Тестовая ситуация: проверка на корректность вводимого числа  $k$ .

Исходные данные:  $p = 31$ ,  $k = 26$  ( $p-1$  не взаимно просто с  $k$ ).

Полученный результат: ошибка о том, числа не взаимно просты.



Ошибка: P - 1 и K не взаимнопростые

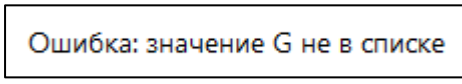
Рисунок 2 – Полученный результат. Тест 2

### Тест 3

Тестовая ситуация: проверка на корректность выбора  $g$ .

Исходные данные:  $p = 41$ ,  $k = 2$ . Неверный  $g$  (не первообразный корень).

Полученный результат: ошибка о том, не находится в списке.



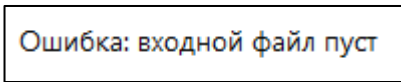
Ошибка: значение G не в списке

Рисунок 3 – Полученный результат. Тест 3

### Тест 4

Тестовая ситуация: проверка на корректность работы программы для пустого (имеет нулевую длину) файла или файл не существует.

Полученный результат: ошибка о том, что файл пуст.



Ошибка: входной файл пуст

Рисунок 4 – Полученный результат. Тест 4

## Тест 5

Тестовая ситуация: проверка на корректность работы программы для маленького размера текстового файла.

Исходные данные:

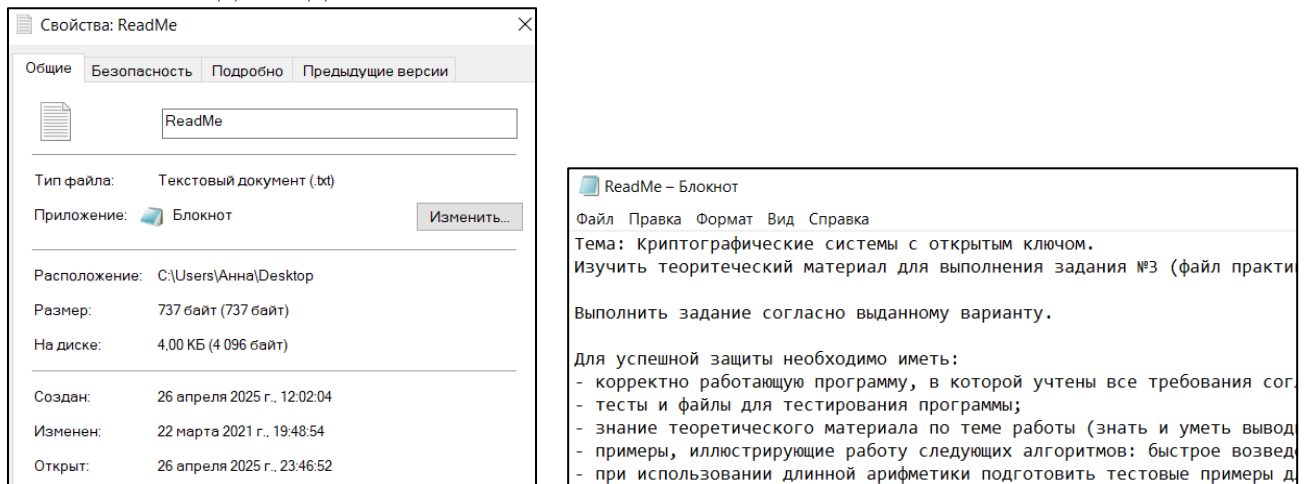


Рисунок 5 – Свойства и содержимое исходного файла. Тест 5

**Математическое ограничение криптосистемы Эль-Гамала - использование простого числа  $p$ , которое больше 255 (максимальное значение байта).**

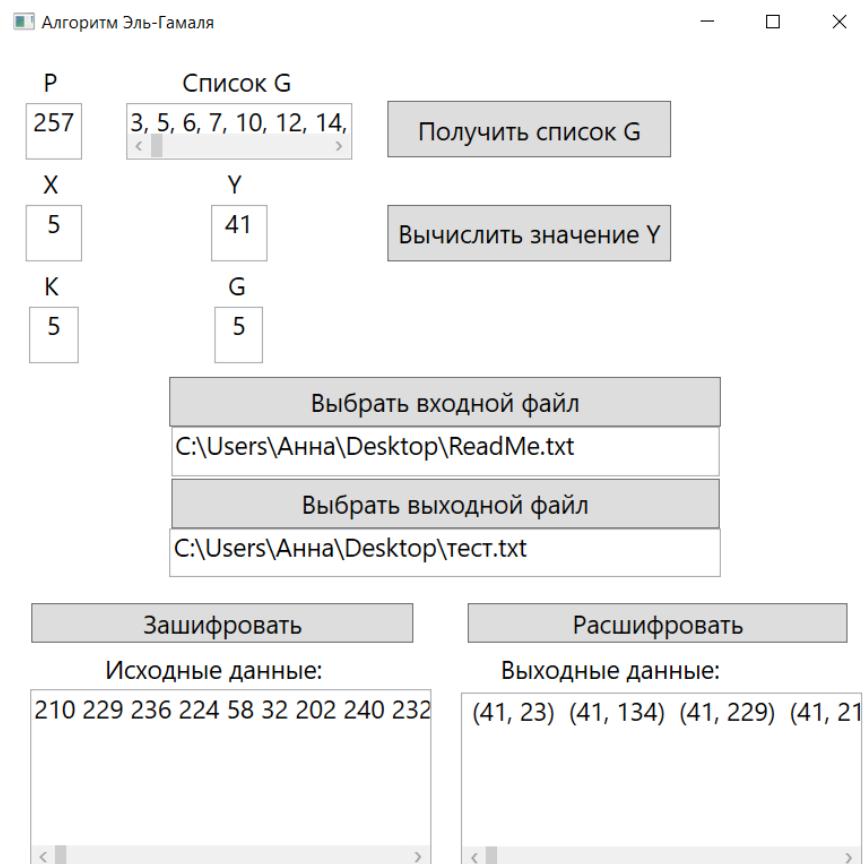


Рисунок 6 – Шифрация. Тест 5

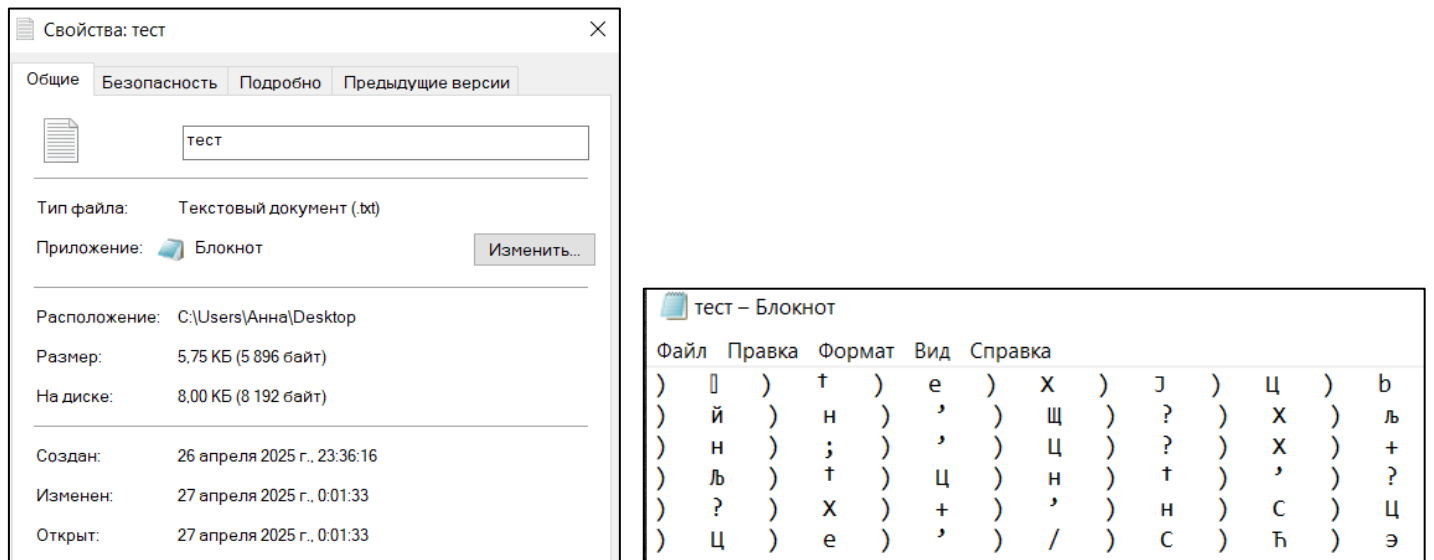


Рисунок 7 – Свойства и содержимое зашифрованного файла. Тест 5

Для шифрования каждый байт преобразуется в два целых числа (a, b), зашифрованный файл должен быть примерно в 8 раз больше исходного (поскольку каждый исходный байт становится двумя 4-байтовыми целыми числами).

Исходные данные:	Выходные данные:
(41, 23) (41, 134) (41, 229) (41, 21)	210 229 236 224 58 32 202 240 232

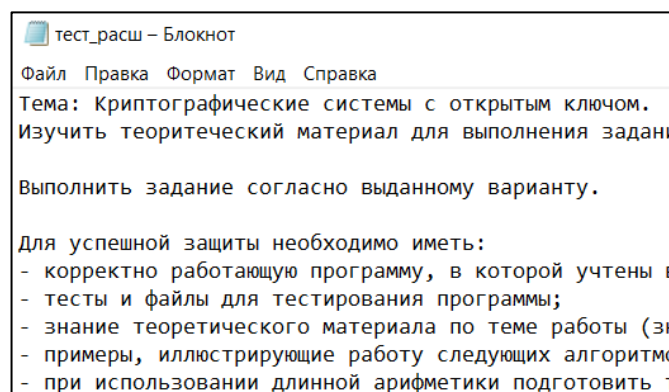


Рисунок 8 – Дешифрация. Тест 5

## Тест 6

Тестовая ситуация: проверка на корректность работы программы для большого размера текстового файла.

Исходные данные:

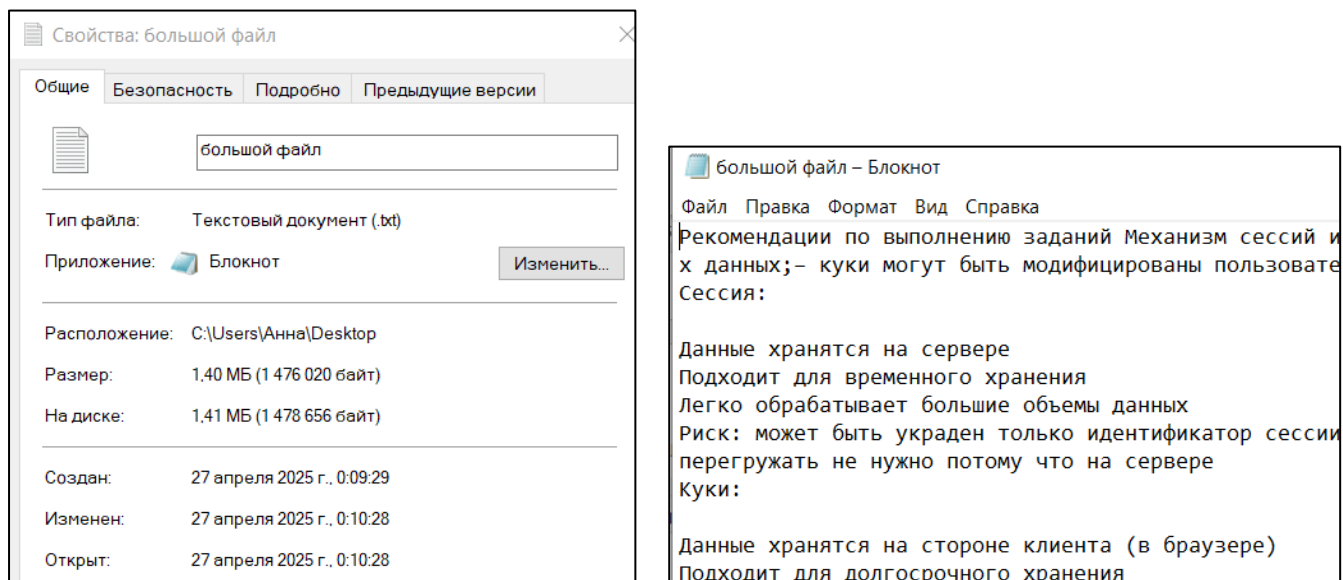


Рисунок 9 – Свойства и содержимое исходного файла. Тест 6

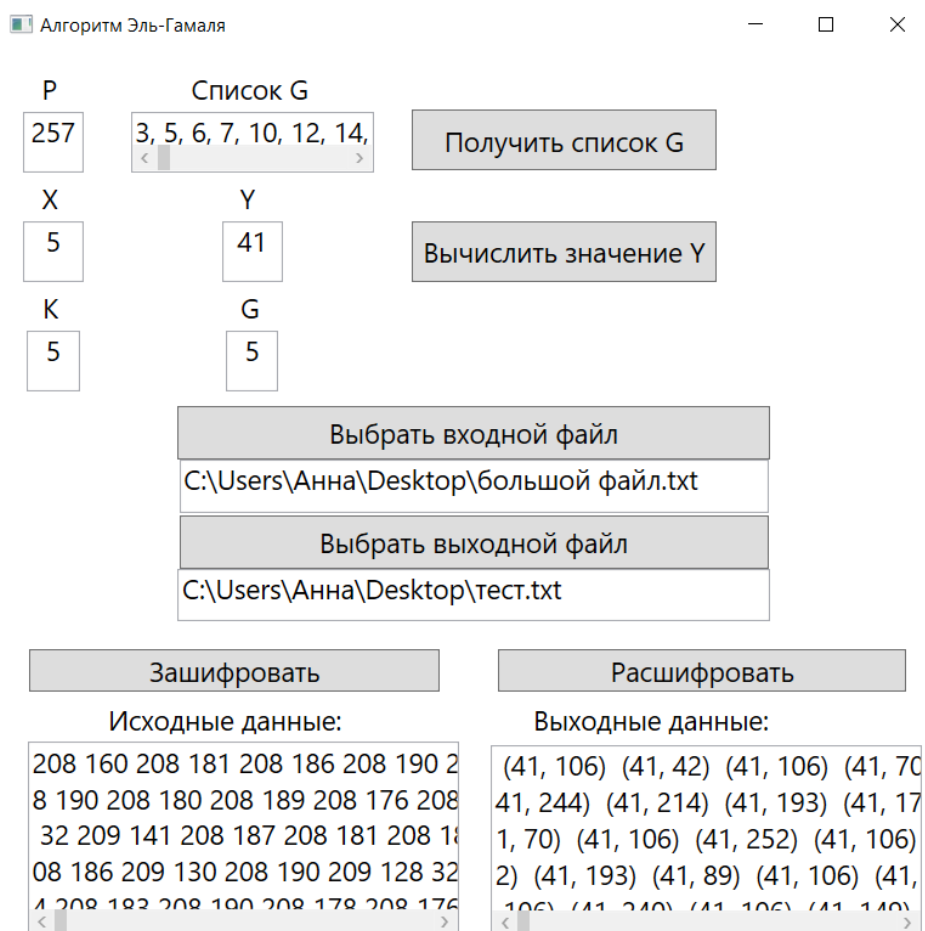


Рисунок 10 – Шифрация. Тест 6

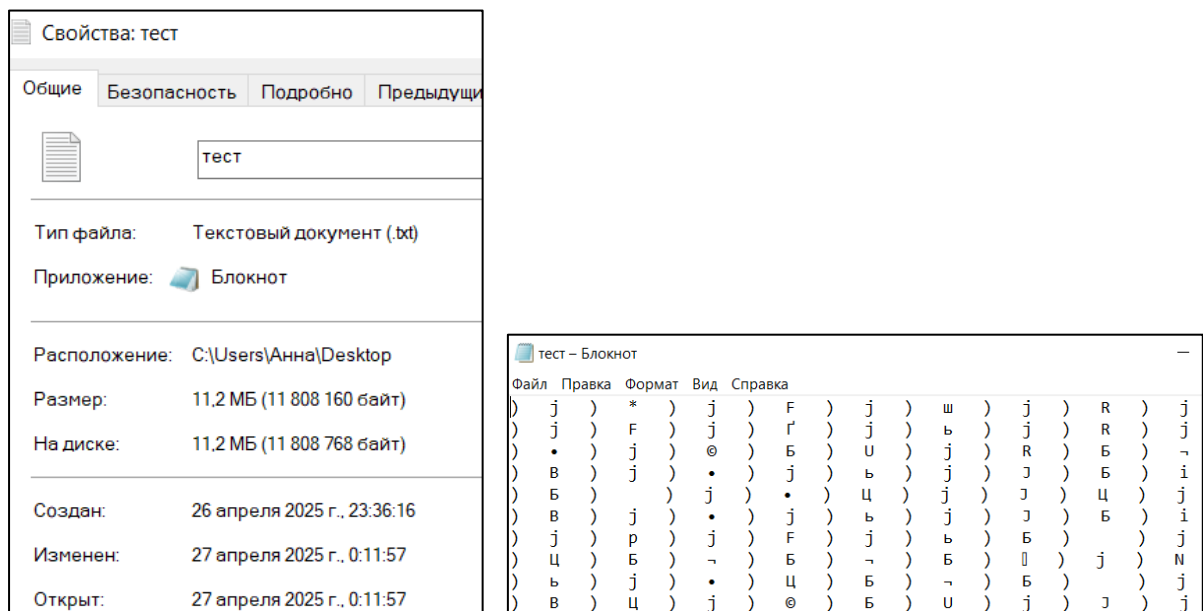


Рисунок 11 – Свойства и содержимое зашифрованного файла. Тест 6

Исходные данные:	Выходные данные:
(41, 106) (41, 42) (41, 106) (41, 70)	208 160 208 181 208 186 208 190 2
41, 244) (41, 214) (41, 193) (41, 17	8 190 208 180 208 189 208 176 208
1, 70) (41, 106) (41, 252) (41, 106)	32 209 141 208 187 208 181 208 14
2) (41, 193) (41, 89) (41, 106) (41,	08 186 209 130 208 190 209 128 32

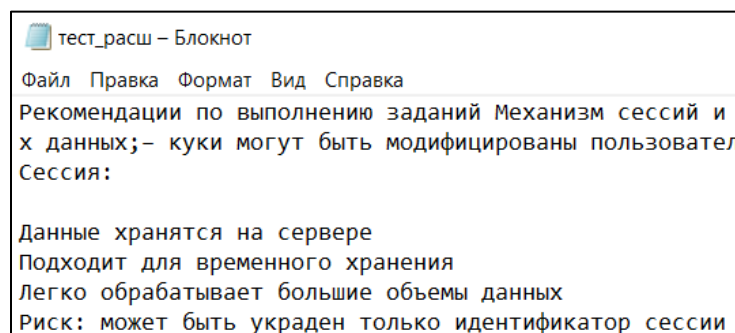


Рисунок 12 – Дешифрация. Тест 6

## Тест 7

Тестовая ситуация: проверка на корректность работы программы для файла музыки.



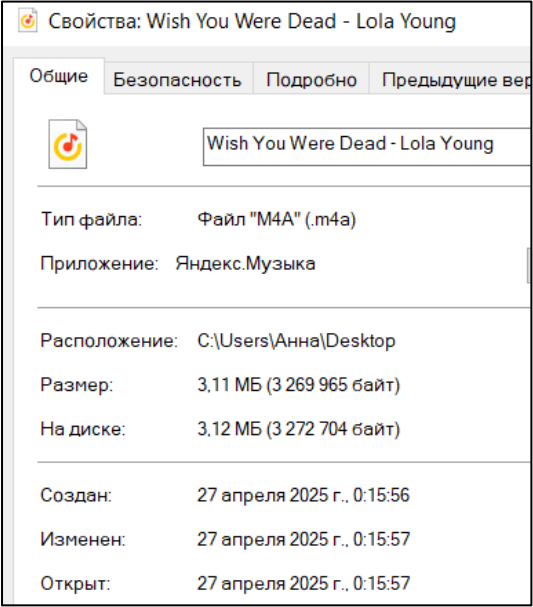


Рисунок 13 – Свойства исходного файла. Тест 7

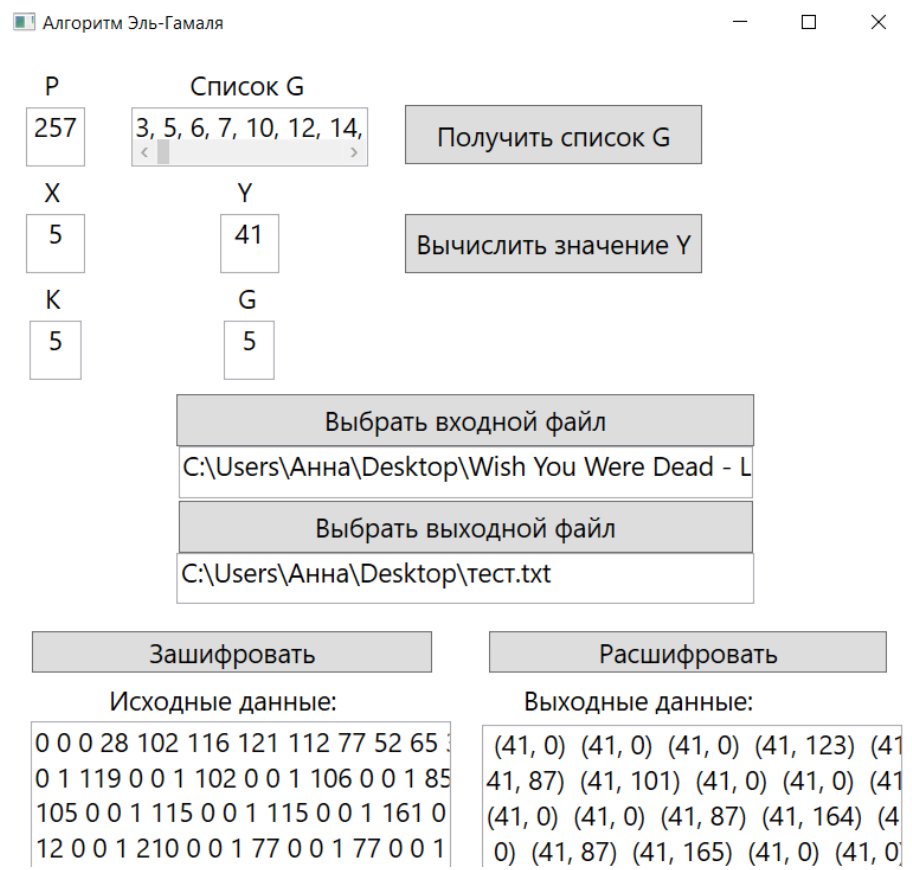


Рисунок 14 – Шифрация. Тест 7

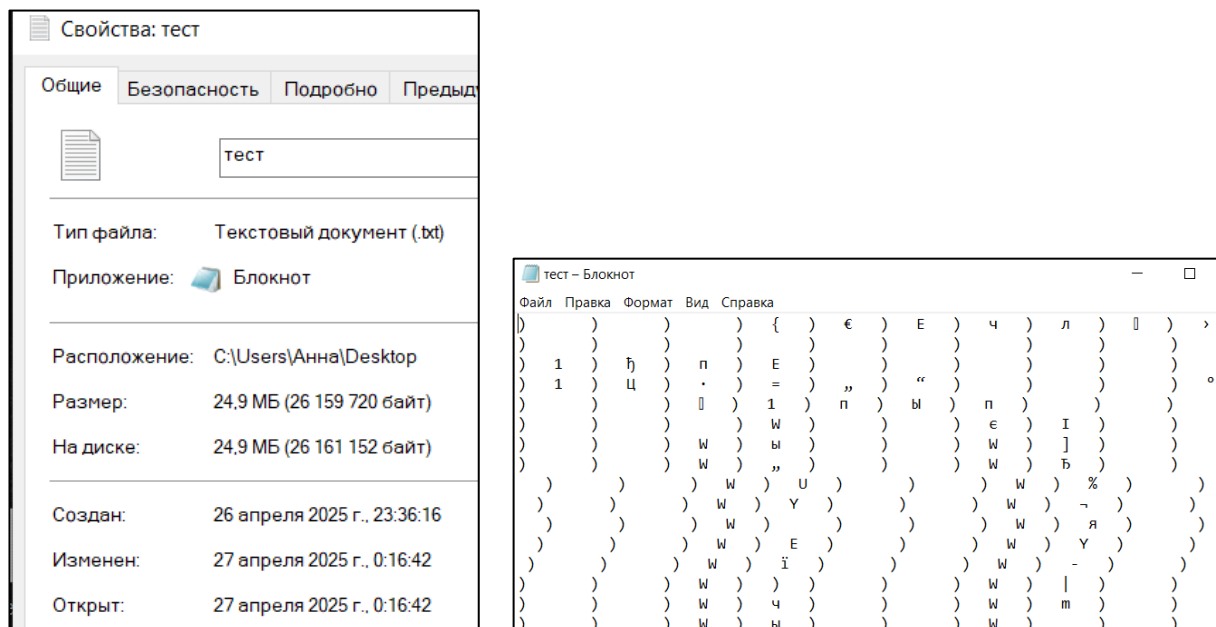


Рисунок 15 – Свойства и содержимое зашифрованного файла. Тест 7

Исходные данные:	Выходные данные:
(41, 0) (41, 0) (41, 0) (41, 123) (41, 87) (41, 101) (41, 0) (41, 0) (41, 0) (41, 0) (41, 87) (41, 164) (41, 0) (41, 87) (41, 165) (41, 0) (41, 0) (41, 148) (41, 0) (41, 0) (41, 87)	0 0 0 28 102 116 121 112 77 52 65 0 1 119 0 0 1 102 0 0 1 106 0 0 1 85 105 0 0 1 115 0 0 1 115 0 0 1 161 0 12 0 0 1 210 0 0 1 77 0 0 1 77 0 0 1 0 1 00 0 0 1 102 0 0 1 120 0 0 1 01

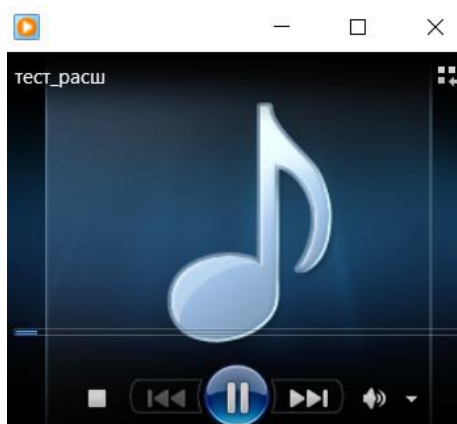


Рисунок 16 – Дешифрация. Тест 7

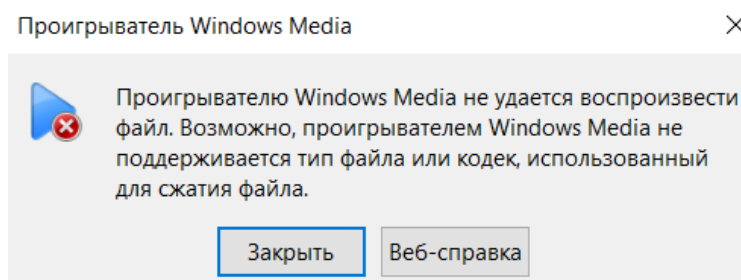


Рисунок 17 – Дешифрация при неправильных вводимых данных. Тест 7

Тест 8

Тестовая ситуация: Проверка на корректность работы программы для файла формата изображения.

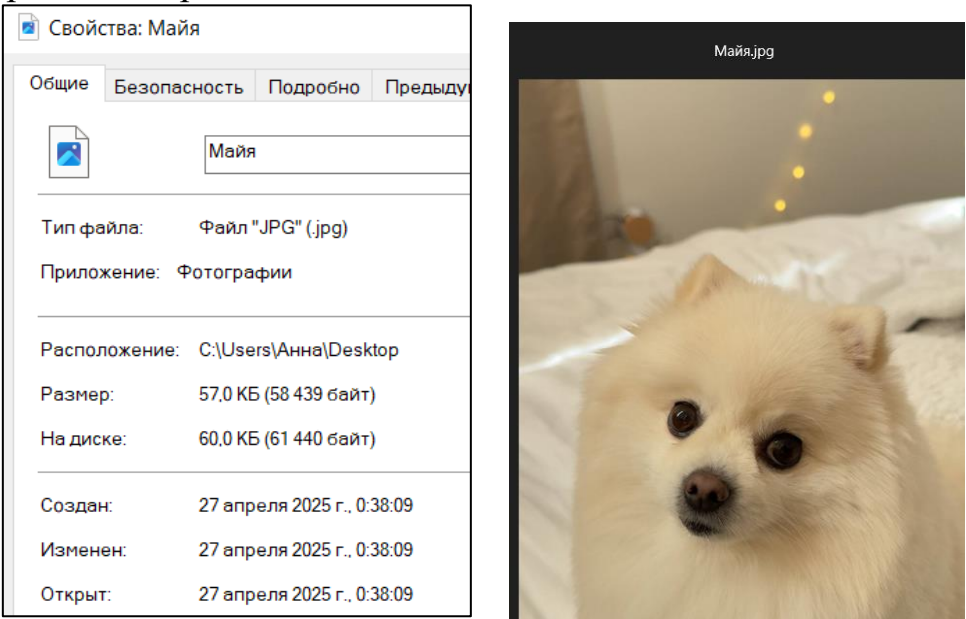


Рисунок 18 – Свойства и содержимое исходного файла. Тест 8

Вводимые данные такие же, как и в предыдущих тестах.

Исходные данные:	Выходные данные:
255 216 255 224 0 16 74 70 73 70 0 0 34 220 141 50 54 205 173 50 141 151 59 143 127 215 249 95 123 55 0 203 112 153 222 73 158 145 121	(41, 83) (41, 31) (41, 83) (41, 213) (41, 0) (41, 2) (41, 42) (41, 50) (41, (41, 127) (41, 100) (41, 67) (41, 2 84) (41, 238) (41, 74) (41, 32) (41,

Рисунок 19 – Шифрация. Тест 8

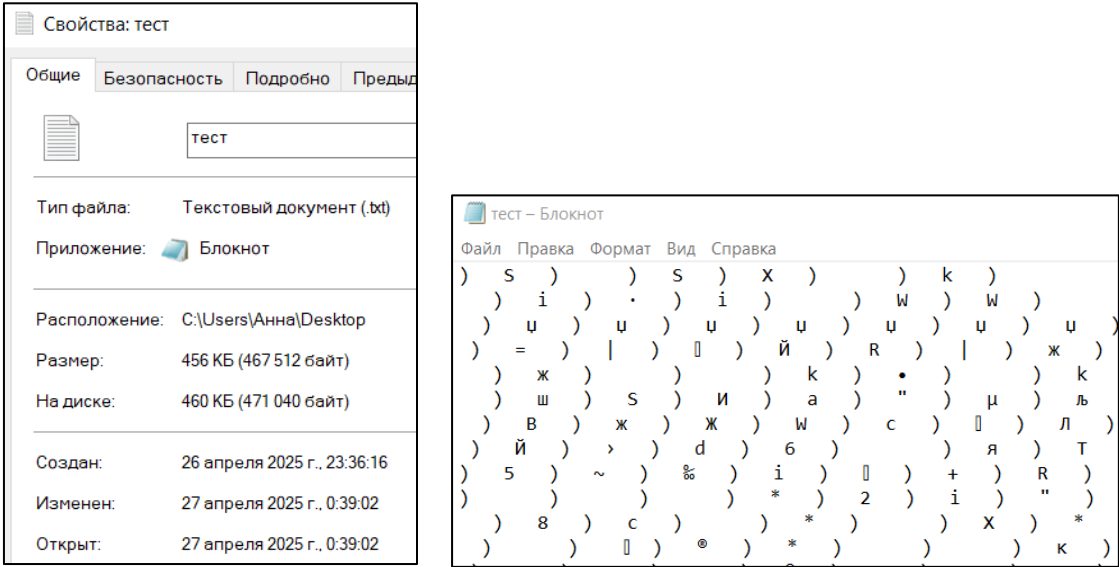


Рисунок 20 – Свойства и содержимое зашифрованного файла. Тест 8

Исходные данные:	Выходные данные:
(41, 83) (41, 31) (41, 83) (41, 213)	255 216 255 224 0 16 74 70 73 70 0
(41, 0) (41, 2) (41, 42) (41, 50) (41, 100)	0 34 220 141 50 54 205 173 50 141
(41, 127) (41, 100) (41, 67) (41, 213)	151 59 143 127 215 249 95 123 55 1
84) (41, 238) (41, 74) (41, 32) (41, 100)	0 203 112 153 222 73 158 145 121 1

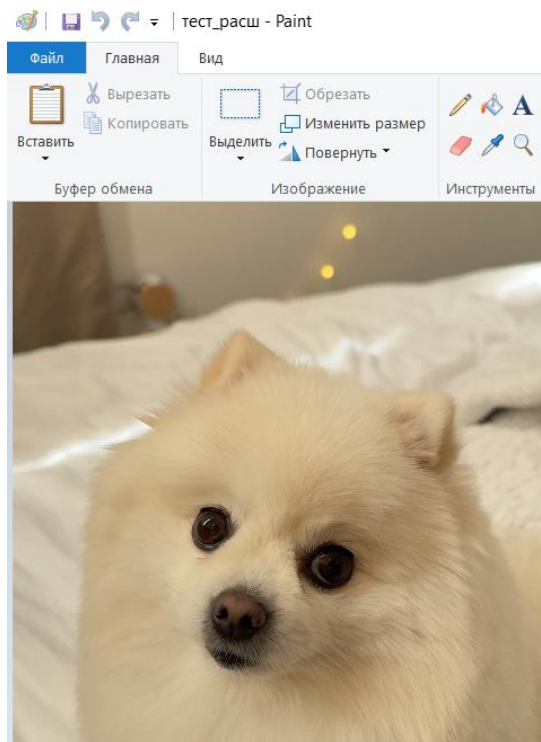


Рисунок 21 – Дешифрация. Тест 8

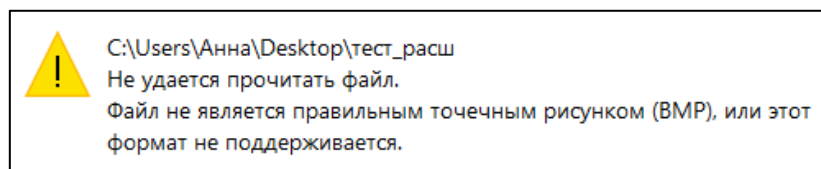


Рисунок 22 – Дешифрация при неправильных вводимых данных. Тест 8

## Тест 9

Тестовая ситуация: проверка на корректность работы программы для файла формата видео.

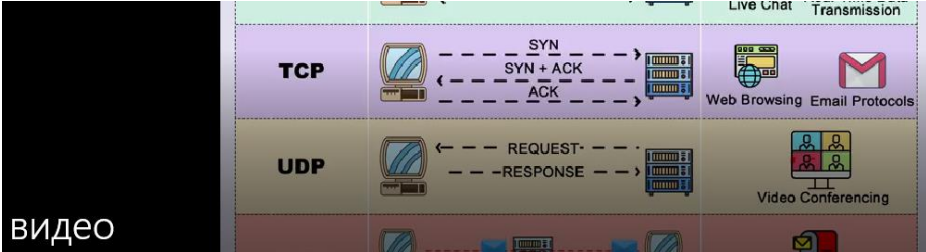
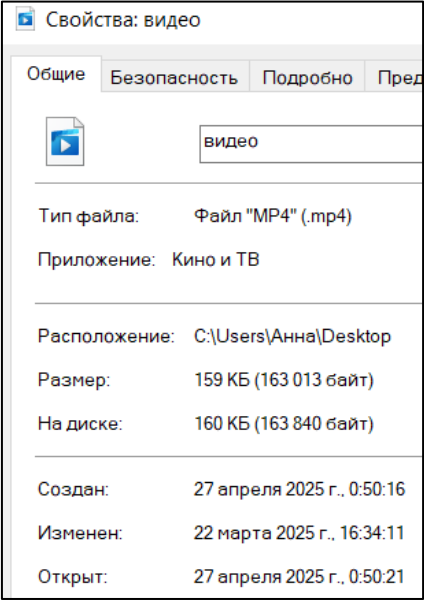


Рисунок 23 – Свойства и содержимое исходного файла. Тест 9

Вводимые данные такие же, как и в предыдущих тестах.

Исходные данные:	Выходные данные:
0 0 0 28 102 116 121 112 109 112 5	(41, 0) (41, 0) (41, 0) (41, 123) (41, 0)
05 110 116 61 50 53 48 32 107 101	83) (41, 245) (41, 64) (41, 0) (41, 0)
20 82 243 79 9 17 32 230 131 184 8	(41, 85) (41, 85) (41, 131) (41, 0) (41, 0)
161 204 33 136 159 53 132 1 184 2	(41, 0) (41, 0) (41, 0) (41, 0) (41, 2)

Рисунок 24 – Шифрация. Тест 9

Рисунок 25 – Содержимое зашифрованного файла. Тест 9

Исходные данные:	Выходные данные:
(41, 0) (41, 0) (41, 0) (41, 123) (41, 83) (41, 245) (41, 64) (41, 0) (41, 85) (41, 85) (41, 131) (41, 0) (41, 0) (41, 0) (41, 0) (41, 2	0 0 0 28 102 116 121 112 109 112 5 05 110 116 61 50 53 48 32 107 101 20 82 243 79 9 17 32 230 131 184 8 161 204 33 136 159 53 132 1 184 2

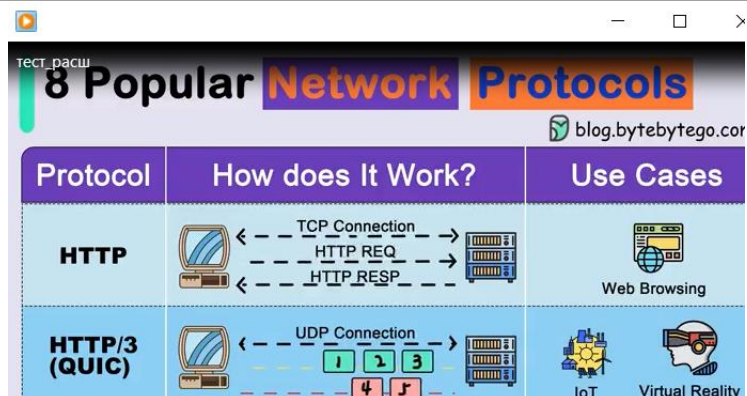


Рисунок 26 – Дешифрация. Тест 9

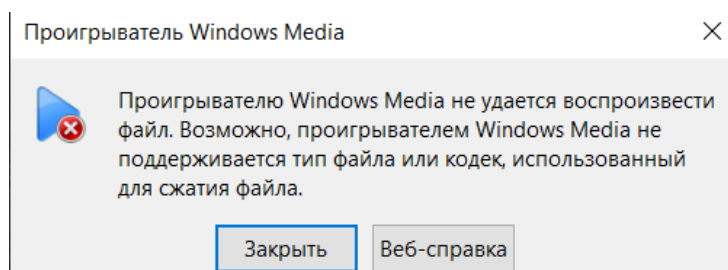


Рисунок 27 – Дешифрация при неправильных вводимых данных. Тест 9

### 3 ПРИМЕР РАБОТЫ АЛГОРИТМА

#### 1. Работа алгоритма быстрого возведения в степень

$$x = a^z \bmod m$$

$a = 17, z = 45, m = 53$ , где  $a$  – основание степени,  $z$  – степень,  $x$  – результат,  $i$  – шаги.

$a$	$z$	$x$	$i$
17	45	1	0
17	44	$1 \times 17 = 17$	1
$17 * 17 = 289 \bmod 53 = 24$	22	17	2
$24 * 24 = 576 \bmod 53 = 46$	11	17	3
46	10	$17 * 46 = 782 \bmod 53$ $= 782 - 14 * 53 = 40$	4
$46 * 46 = 2116 \bmod 53 = 49$	5	40	5
49	4	$40 * 49 = 1960 \bmod 53$ $= 1960 - 36 * 53 = 52$	6
$49 * 49 = 2401 \bmod 53 = 16$	2	52	7
$16 * 16 = 256 \bmod 53 = 44$	1	52	8
44	0	$52 * 44 = 2288 \bmod 53$ $= 2288 - 43 * 53 = 9$	9

$$17^{45} \bmod 53 = 9$$

#### 2. Поиска всех первообразных корней при $p = 61$

Условие для первообразного корня:

$$(g^{\varphi(p)} = 1 \bmod p) \&\& (g^l \neq 1 \bmod p; 1 \leq l \leq \varphi(p))$$

Пусть  $p - 1 = 60$ . Простые делители  $p - 1 = 60 = 2^2 \cdot 3 \cdot 5 = \{q_0 = 2, q_1 = 3, q_2 = 5\}$ .  $l = \frac{p-1}{q_i} = \{30, 20, 12\}$

#### Проверка на первообразность

$g_i$	$g^{30} \neq 1 \bmod 61$	$g^{20} \neq 1 \bmod 61$	$g^{12} \neq 1 \bmod 61$
1	–	–	–
2	60	47	9
3	1	–	–
4	1	–	–
5	1	–	–
6	60	47	20
7	60	47	34
8	60	1	–
9	1	–	–

<b>10</b>	<b>60</b>	<b>13</b>	<b>58</b>
11	60	1	–
12	1	–	–
13	1	–	–
14	1	–	–
15	1	–	–
16	1	–	–
<b>17</b>	<b>60</b>	<b>13</b>	<b>20</b>
<b>18</b>	<b>60</b>	<b>47</b>	<b>58</b>
19	1	–	–
20	1	–	–
21	60	47	1
22	1	–	–
23	60	1	–
24	60	1	–
25	1	–	–
<b>26</b>	<b>60</b>	<b>13</b>	<b>9</b>
27	1	–	–
28	60	1	–
29	60	13	1
<b>30</b>	<b>60</b>	<b>13</b>	<b>34</b>
<b>31</b>	<b>60</b>	<b>13</b>	<b>34</b>
32	60	13	1
33	60	1	–
34	1	–	–
<b>35</b>	<b>60</b>	<b>13</b>	<b>9</b>
36	1	–	–
37	60	1	–
38	60	1	–
39	1	–	–
40	60	47	1
41	1	–	–
42	1	–	–
<b>43</b>	<b>60</b>	<b>47</b>	<b>58</b>
<b>44</b>	<b>60</b>	<b>13</b>	<b>20</b>
45	1	–	–
46	1	–	–
47	1	–	–
48	1	–	–
49	1	–	–
50	60	1	–
<b>51</b>	<b>60</b>	<b>13</b>	<b>58</b>
52	1	–	–
53	60	1	–



54	60	47	34
55	60	47	20
56	1	–	–
57	1	–	–
58	1	–	–
59	60	47	9
60	1	–	–

Количество первообразных корней должно равняться  $\phi(p-1) = \phi(60) = \phi(22)\phi(3)\phi(5) = 2 * 2 * 4 = 16$ .

Множество первообразных корней для  $p = 61 \Rightarrow \{2, 6, 7, 10, 17, 18, 26, 30, 31, 35, 43, 44, 51, 54, 55, 59\}$

### 3. Работа расширенного алгоритма Евклида

$$Xi * a + Yi * b = \text{НОД}(a, b) = 1$$

Пусть  $a = 713, b = 256$

i	q	a <sub>0</sub>	a <sub>1</sub>	x <sub>0</sub>	x <sub>1</sub>	y <sub>0</sub>	y <sub>1</sub>
0	-	256	713	1	0	0	1
1	0	713	256	0	1	1	0
2	2	256	201	1	-2	0	1
3	1	201	55	-2	3	1	-1
4	3	55	36	3	-11	-1	4
5	1	36	19	-11	14	4	-5
6	1	19	17	14	-25	-5	9
7	1	17	2	-25	39	9	-14
8	8	2	1	39	-337	-14	121
9	2	1	0	-337	713	121	-256

$x_1 = -337, y_1 = 121$

$$-337 * 256 + 121 * 713 = 1$$