

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический
университет Петра Великого»

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Направление: 02.03.01 «Математика и компьютерные науки»

Обработка файлов в Haskell
Практическое задание 3

Студент,
группы 5130201/20101

_____ Астафьев И. Е.

Преподаватель

_____ Моторин Д. Е.

«_____» _____ 2024г.

Санкт-Петербург, 2024

Содержание

Введение	3
1 Подготовка файла	4
2 Теоретические сведения	4
2.1 Шифр Цезаря	4
2.2 Стеганография	5
3 Реализация программы	5
3.1 Шифрование текста	5
3.2 Кодировка текста в изображение	6
3.3 Декодировка текста из файла	6
3.4 Вызов функций	6
4 Результаты	7
5 Заключение	10
Приложение А. Код программы	11
Приложение Б. Тексты	15

Введение

В данном отчете описаны результаты выполнения практического задания по кодированию текстового файла в изображение с использованием шифра Цезаря с использованием функционального языка программирования Haskell и инструмента сборки Stack.

Постановка задач:

1. Найти портрет указанного человека: **Циолковский, Константин Эдуардович**. Перевести изображение в формат .bmp (24-разрядный), при необходимости изменить ширину и высоту изображения без искажений. Сохранить в файл формата .txt фрагмент биографии (не менее 1000 символов без пробелов, текст не должен обрываться на середине слова или предложения). Закодировать текст в изображение методом **шифр Цезаря** (смещение задается пользователем). Ключ к шифру записывается в имя файла. Написать функцию расшифровывающую текст из изображения используя ключ из имени файла и сохраняющую результат в отдельный текстовый файл. Создать функции, шифрующие текст в последний бит каждого байта, последние два бита каждого байта, ..., все биты в байте. В отчете привести примеры искажений изображения.
2. Создать проект в stack. Все чистые функции записать в библиотеку Lib.hs и ограничить доступ к вспомогательным функциям. Использовать do-нотацию для работы с внешними файлами.

1 Подготовка файла

Для начала требуется найти портрет Константина Эдуардовича Циолковского. Было найдено изображение формата jpg, далее оно было переведено в 24-разрядный формат bmp через конвертер в Интернете Рис. 1.



Рис. 1. Портрет Циолковского К. А.

Изображение помещено в файл «ciol.bmp», который занимает на диске 157 Кб.

Далее английский текст с фрагментом биографии ученого был помещен в файл «bio.txt». Текст с подсчетом количества символов без пробелов представлен на Рис. 2. Как можно увидеть текст содержит 7773 символов.

Tsiolkovsky was born in Izhevskoye (now in Spassky District, Ryazan Oblast), in the Russian Empire, to a middle-class family. His father was a Polish forester of Roman Catholic faith who relocated to Russia; his Russian Orthodox mother Maria Ivanovna Yumasheva was of mixed Volga Tatar and Russian origin. His father was successively a forester, teacher, and minor government official. At the age of 9, Konstantin caught scarlet fever and lost his hearing. When he was 13, his mother died. He was not admitted to elementary schools because of his hearing problem, so he was self-taught. As a reclusive home-schooled child, he passed much of his time by reading books and became interested in mathematics and physics. As a teenager, he began to contemplate the possibility of space travel. Tsiolkovsky spent three years attending a Moscow library, where Russian cosmism proponent Nikolai Fyodorov worked. He later came to believe that colonizing space would lead to the perfection of the human species, with immortality and a carefree existence. Additionally, inspired by the fiction of Jules Verne, Tsiolkovsky theorized many aspects of space travel and rocket propulsion. He is considered the father of spaceflight and the first person to conceive the space elevator, becoming inspired in 1895 by the newly constructed Eiffel Tower in Paris. Despite the youth's growing knowledge of physics, his father was concerned that he would not be able to provide for himself financially as an adult and brought him

Всего символов	Без пробелов	Кол-во слов
7773	6518	1256

Проверить SEO-данные

Рис. 2. Количество символов в тексте

Текстовый файл занимает на диске 8 Кб.

2 Теоретические сведения

2.1 Шифр Цезаря

Шифр Цезаря — это один из самых простых и известных методов шифрования текста. Он относится к классическим шифрам подстановки, где каждая буква заменя-

ется на другую, расположенную на фиксированное число позиций вперед или назад в алфавите.

Название шифра связано с римским полководцем и политиком Гаем Юлием Цезарем, который, как считается, использовал этот метод для секретной переписки. Например, при сдвиге на три позиции буква «А» заменяется на «D», «В» на «Е» и так далее. Когда алфавит заканчивается, сдвиг продолжается с начала (циклический переход).

Формула шифрования:

$$C = (P + K) \mod N$$

Формула дешифрования:

$$P = (C - K + N) \mod N$$

где: P — индекс символа в алфавите, K — величина сдвига, N — количество символов в алфавите, C — индекс зашифрованного символа.

2.2 Стеганография

Стеганография — это метод скрытия информации, при котором сама передача скрытых данных остается незаметной. В отличие от криптографии, которая маскирует содержимое сообщения, стеганография скрывает само его существование. Термин происходит от греческих слов «steganos» (скрытый) и «graphy» (письмо).

Идея стеганографии заключается в встраивании секретных данных в несекретные носители, такие как изображения, аудиофайлы, видео или текст. Основной задачей является изменение носителя таким образом, чтобы изменения были незаметны для человеческого восприятия или анализа.

В изображениях используется изменение младших значащих битов (LSB, Least Significant Bit) пикселей. Например, если каждый цветовой компонент (красный, зеленый, синий) представлен байтом, замена последнего бита каждого из них позволяет встраивать данные с минимальным визуальным воздействием.

В данной работе заменяются n последних битов в каждом байте изображения на биты текста.

3 Реализация программы

Полный исходный код представлен в [Приложение А. Код программы](#).

3.1 Шифрование текста

Функция `caesarCipher` выполняет шифрование текста с использованием шифра Цезаря, сдвигая каждую букву на заданное количество позиций, указанное ключом. Она принимает на вход текст типа `T.Text` и целое число `key`, определяющее сдвиг. Для каждого символа проверяется, является ли он буквой в диапазоне от `a` до `z` или от `A` до `Z`. Если символ является буквой, его позиция в алфавите изменяется по формуле (текущая позиция - позиция начала алфавита + ключ) % 26 + позиция начала алфавита. Если символ не является буквой, он остается неизменным.

Функция `caesarDecipher` выполняет расшифровку текста, зашифрованного шифром Цезаря, сдвигая буквы в обратном направлении. Она принимает тот же набор параметров и вызывает функцию `caesarCipher` с отрицательным значением ключа `-key`. Эти функции подходят для выполнения базового шифрования и дешифрования текста с использованием простого метода шифра Цезаря.

3.2 Кодировка текста в изображение

Функция `byteToBits` преобразует байт в список из восьми битов, где каждый бит представлен как 1 или 0. Биты вычисляются с помощью функции `testBit`, которая проверяет установлен ли бит в заданной позиции. Функция `replaceIBit` заменяет бит в указанной позиции байта на новый бит (0 или 1). Если бит не равен 0 или 1, генерируется ошибка. Функция `chunksOf` разбивает список на части заданной длины, возвращая список списков.

Функция `replaceBitsInByteString` заменяет определенное количество младших битов каждого байта в исходной строке байтов `original` на биты из строки `replacing`. Проверяется, достаточно ли бит в исходной строке для замены. Биты из строки `replacing` разбиваются на группы по заданной длине и последовательно заменяются в байтах исходной строки с помощью вспомогательной функции `replaceNthBits`.

Функция `encodeTextToImage` кодирует текст из текстового файла в изображение, используя шифр Цезаря для предварительной обработки текста. Сначала она считывает изображение и текстовый файл, затем шифрует текст с помощью функции `caesarCipher`. Затем текст преобразуется в байтовый формат и записывается в изображение, заменяя младшие биты каждого пикселя. Файл изображения сохраняется с разными вариантами замены битов (от 1 до 8), а результат записывается в новые файлы с измененным содержимым изображения. Таким образом, после исполнения данной функции создается 8 новых файлов `.bmr`, названия которых сделаны по следующему шаблону:

`<сдвиг шифра цезаря>_<количество заменяемых битов>_<длина текста>_<имя исходного изображения>.bmr`

3.3 Декодировка текста из файла

Функция `extractBitsFromByte` извлекает младшие биты из байта, начиная с самого младшего и до заданного числа `n`. Биты возвращаются в виде списка чисел 1 или 0. Функция `bitsToBytes` преобразует список битов в строку байтов. Она группирует биты по восемь, собирает их в байты с помощью побитовых операций, а затем рекурсивно обрабатывает оставшиеся биты.

Функция `decodeTextFromImage` выполняет извлечение закодированного текста из изображения. Из имени файла изображения она считывает параметры кодировки: сдвиг шифра Цезаря, количество битов, используемых для кодировки, и длину текста. Затем она извлекает содержимое изображения в формате RGBA, извлекает нужное количество битов с использованием функции `extractBitsFromByte` и преобразует их в байты с помощью `bitsToBytes`. После этого байты расшифровываются с использованием шифра Цезаря. Результирующий текст сохраняется в новый файл с именем, основанным на исходном имени изображения.

Функция `decodeTextsFromImages` выполняет декодирование текста из списка изображений. Она вызывает `decodeTextFromImage` для каждого файла в списке, извлекая и расшифровывая закодированный текст по указанным параметрам.

3.4 Вызов функций

Функция `main` является основной точкой входа программы и реализует процесс кодирования текста в изображение с последующим декодированием.

Сначала программа запрашивает у пользователя величину сдвига для шифра Цезаря и считывает её с клавиатуры. Эта величина преобразуется в целое число, которое используется для шифрования текста. Затем задаются имена исходного изображения (`imageFileName`) и файла с текстом (`textFileName`), которые участвуют в процессе.

С помощью функции `encodeTextToImage` текст из файла кодируется в изображение. Эта функция принимает сдвиг для шифра Цезаря, шифрует текст и встраивает его в изображение, используя разные значения числа битов для кодирования.

После кодирования программа читает исходный текст из файла и формирует список имён изображений, созданных на этапе кодирования. Затем вызывается функция `decodeTextsFromImages`, которая последовательно извлекает закодированный текст из каждого изображения, декодирует его и сохраняет в новые файлы.

Таким образом, `main` связывает все основные функции программы, обеспечивая как процесс кодирования текста в изображение, так и его последующее декодирование.

4 Результаты

Исходный текст, а также зашифрованный шифром Цезаря текст, который закодирован в изображение, представлены в [Приложение Б. Тексты](#).

При запуске программы пользователю необходимо ввести сдвиг шифра Цезаря (Рис. 3).

```
PS D:\Haskell\projects\picture-cipher> stack exec picture-cipher-exe
enter Caesar shift:
5
```

Рис. 3. Ввод сдвига для шифра Цезаря

Затем создается 17 новых файлов (Рис. 4): текстовый файл с закодированным текстом, 8 изображений с зашифрованным в них текстом, а также 8 расшифрованных текстов. Как можно увидеть, в названиях файлов содержится метаданная для декодирования. Также пользователю выводится соответствующее сообщение (Рис. 5).

















 5_1_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_1_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_2_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_2_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_3_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_3_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_4_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_4_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_5_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_5_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_6_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_6_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_7_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_7_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ
 5_8_7786_ciol2.bmp	28.11.2024 1:00	Файл "BMP"	209 КБ
 5_8_7786_ciol2.bmp_decoded.txt	28.11.2024 1:00	Текстовый докум...	8 КБ

Рис. 4. Созданные файлы

```
converted
Decoded text written to 5_1_7786_ciol2.bmp_decoded.txt
Decoded text written to 5_2_7786_ciol2.bmp_decoded.txt
Decoded text written to 5_3_7786_ciol2.bmp_decoded.txt
Decoded text written to 5_4_7786_ciol2.bmp_decoded.txt
Decoded text written to 5_5_7786_ciol2.bmp_decoded.txt
```

Рис. 5. Вывод сообщения для пользователя

Как уже было указано ранее на Рис. 1 - оригинальное изображение с портретом Циолковского.

На Рис. 6 - 13 представлены изображения с закодированным в них текстом для $n = 1..8$.



Рис. 6. Изображение для $n = 1$



Рис. 7. Изображение для $n = 2$



Рис. 8. Изображение для $n = 3$



Рис. 9. Изображение для $n = 4$



Рис. 10. Изображение для $n = 5$



Рис. 11. Изображение для $n = 6$



Рис. 12. Изображение для $n = 7$



Рис. 13. Изображение для $n = 8$

Как можно заметить, для $n = 1, 2$ (Рис. 6, 7) никаких изменений на первый взгляд не видно, для $n = 3, 4$ (Рис. 8, 9) заметны легкие искажения в верхней части изображений, для $n = 5, 6$ (Рис. 10, 11) отчетливо видны искажения цвета, для $n = 7, 8$ (Рис. 12, 13) часть изображения настолько искажена, что в этих участках исходное изображение совершенно неразличимо (не видны очертания).

5 Заключение

В результате выполнения практического задания №3, был создан проект в `stack`, в котором реализована логика шифрования текста в изображение с предварительной кодировкой шифром Цезаря, и расшифровка по ключу из названия закодированного файла. Готовая программа:

- Шифрует полученный текстовый файл в изображение.
- Сохраняет несколько новых измененных изображений.
- Декодирует зашифрованные тексты из изображений.
- Сохраняет файлы с расшифрованным текстом.

Приложение А. Код программы

Lib.hs

```
1 module Lib
2   ( encodeTextToImage
3   , decodeTextsFromImages
4   ) where
5
6 import qualified Data.Text as T
7 import Data.Char (ord, chr)
8 import qualified Data.ByteString as B
9 import qualified Data.Text.Encoding as TE
10 import Data.Bits (testBit, setBit, clearBit)
11 import Data.Word (Word8)
12 import Codec.BMP
13
14 caesarCipher :: T.Text -> Int -> T.Text
15 caesarCipher str key = T.map shiftChar str
16   where
17     shiftChar c
18       | c >= 'a' && c <= 'z' = chr $ (ord c - ord 'a' + key) `mod` 26 + ord 'a'
19       | c >= 'A' && c <= 'Z' = chr $ (ord c - ord 'A' + key) `mod` 26 + ord 'A'
20       | otherwise = c
21
22 caesarDecipher :: T.Text -> Int -> T.Text
23 caesarDecipher str key = caesarCipher str (-key)
24
25 -- переводит байт в список битов
26 byteToBits :: Word8 -> [Int]
27 byteToBits byte = [if testBit byte i then 1 else 0 | i <- [7,6..0]]
28
29 -- Заменяет i-й бит в байте на заданный бит (0 или 1)
30 replaceIBit :: Word8 -> Int -> Int -> Word8
31 replaceIBit byte n bit
32   | bit == 1 = setBit byte n
33   | bit == 0 = clearBit byte n
34   | otherwise = error "Bit must be 0 or 1"
35
36 -- Разбивает список на части длиной k
37 chunksOf :: Int -> [a] -> [[a]]
38 chunksOf _ [] = []
39 chunksOf k xs = take k xs : chunksOf k (drop k xs)
40
41 replaceBitsInByteString :: B.ByteString -> B.ByteString -> Int -> B.ByteString
42 replaceBitsInByteString original replacing n
43   | n < 1 || n > 8 = error "Number of bits to replace must be between 1 and 8"
44   | B.length replacing * 8 > B.length original * n = error "Not enough bits in"
45   ↪ original"
46   | otherwise = B.pack $ go (B.unpack original) (chunksOf n replacingBits)
```

```

46 where
47     -- Преобразуем байты из replacing в список битов
48     replacingBits = concatMap byteToBits (B.unpack replacing)
49
50     -- Рекурсивная функция для замены битов и сохранения неизменных байтов
51     go [] [] = [] -- Все байты обработаны
52     go (o:os) [] = o : go os [] -- Добавляем оставшиеся байты из original, если нет
53     -- замены
54     go (o:os) (r:rs) =
55         let modifiedByte = replaceNthBits o r -- Заменяем последние n бит
56         in modifiedByte : go os rs -- Добавляем измененный байт и рекурсивно
57         -- обрабатываем остаток
58
59     -- Заменяет последние n бит байта на новые биты из replacing
60     replaceNthBits :: Word8 -> [Int] -> Word8
61     replaceNthBits byte newBits =
62         foldl (\acc (bit, idx) -> replaceIBit acc idx bit) byte (zip newBits (reverse
63         -- [0..n-1]))
64
65 encodeTextToImage :: FilePath -> FilePath -> Int -> IO ()
66 encodeTextToImage imageFileName textFileName caesarK = do
67     Right bmp <- readBMP imageFileName
68     let rgba = unpackBMPToRGBA32 bmp
69     let (width, height) = bmpDimensions bmp
70     textFile <- B.readFile textFileName
71
72     -- Функция для генерации имени файла
73     let glitchedFileName n = (show caesarK) ++ "_" ++ (show n) ++ "_" ++ (show $
74     -- B.length textFile) ++ "_" ++ imageFileName
75
76     let codedTextFile = TE.encodeUtf8 $ caesarCipher (TE.decodeUtf8 textFile) caesarK
77
78     B.writeFile ("coded_" ++ textFileName) codedTextFile
79
80     -- Итерация по значениям от 1 до 8
81     mapM_ (\n -> do
82         let modifiedRGBA = B.reverse $ replaceBitsInByteString (B.reverse rgba)
83         -- (codedTextFile) n
84         let glitchedBMP = packRGBA32ToBMP width height modifiedRGBA
85         -- print $ B.take 10 $ B.reverse modifiedRGBA
86         writeBMP (glitchedFileName n) glitchedBMP [1..8]
87         -- B.writeFile glitchedFileName $ replaceBitsInByteString imageFile textFile 8
88         putStrLn "converted"
89
90     -- Извлечение последних n бит из каждого байта
91     extractBitsFromByte :: Word8 -> Int -> [Int]
92     extractBitsFromByte byte n = reverse [if testBit byte i then 1 else 0 | i <- [0..n-1]]
93
94     bitsToBytes :: [Int] -> B.ByteString

```

```

91 bitsToBytes [] = B.empty
92 bitsToBytes bits =
93     B.cons (fromIntegral (foldl (\acc (b, i) -> if b == 1 then setBit acc i else acc) (0
94         ↪ :: Word8) (zip (take 8 bits) [7,6..0])))
95         (bitsToBytes (drop 8 bits))
96
97 -- Восстановление текста из изображения
98 decodeTextFromImage :: FilePath -> IO ()
99 decodeTextFromImage imageFileName = do
100     -- Извлекаем параметры из имени файла
101     let [caesarKStr, nStr, textLenStr, _] = T.splitOn (T.pack "_") (T.pack
102         ↪ imageFileName)
103     let caesarK = read (T.unpack caesarKStr) :: Int
104     let n = read (T.unpack nStr) :: Int
105     let textLen = read (T.unpack textLenStr) :: Int
106
107     -- Читаем изображение
108     Right bmp <- readBMP imageFileName
109     let rgba = B.reverse $ unpackBMPToRGBA32 bmp
110
111     -- Извлекаем последние n бит из каждого байта
112     let extractedBits = concatMap (`extractBitsFromByte` n) (B.unpack rgba)
113
114     -- Берем нужное количество бит и преобразуем в байты
115     let textBytes = bitsToBytes $ take (textLen * 8) extractedBits
116
117     -- Декодируем байты в текст
118     let decodedText = caesarDecipher (TE.decodeUtf8 textBytes) caesarK
119
120     -- Записываем восстановленный текст в файл
121     let outputFileName = imageFileName ++ "_decoded.txt"
122     B.writeFile outputFileName $ TE.encodeUtf8 decodedText
123     putStrLn $ "Decoded text written to " ++ outputFileName
124
125 -- Декодирование текста из всех изображений
126 decodeTextsFromImages :: [FilePath] -> IO ()
127 decodeTextsFromImages imageFileNames = mapM_ decodeTextFromImage imageFileNames

```

Main.hs

```

1 module Main (main) where
2
3 import Lib (encodeTextToImage, decodeTextsFromImages)
4 import qualified Data.ByteString as B
5
6 main :: IO ()
7 main = do
8     putStrLn "enter Caesar shift: "
9     caesarShiftInput <- getLine

```

```
10
11   let caesarShift = (read caesarShiftInput :: Int)
12   let imageFileName = "ciol2.bmp"
13   let textFileName = "bio.txt"
14
15   encodeTextToImage imageFileName textFileName caesarShift
16
17   textFile <- B.readFile textFileName
18   decodeTextsFromImages [(show caesarShift) ++ "_" ++ (show x) ++ "_" ++ (show $
↪   B.length textFile) ++ "_" ++ imageFileName | x <- [1..8]]
```

Приложение Б. Тексты

bio.txt

- 1 Tsiolkovsky was born in Izhevskoye (now in Spassky District, Ryazan Oblast), in the
→ Russian Empire, to a middle-class family. His father was a Polish forester of Roman
→ Catholic faith who relocated to Russia; his Russian Orthodox mother Maria Ivanovna
→ Yumasheva was of mixed Volga Tatar and Russian origin. His father was successively a
→ forester, teacher, and minor government official. At the age of 9, Konstantin caught
→ scarlet fever and lost his hearing. When he was 13, his mother died. He was not
→ admitted to elementary schools because of his hearing problem, so he was
→ self-taught. As a reclusive home-schooled child, he passed much of his time by
→ reading books and became interested in mathematics and physics. As a teenager, he
→ began to contemplate the possibility of space travel. Tsiolkovsky spent three years
→ attending a Moscow library, where Russian cosmism proponent Nikolai Fyodorov worked.
→ He later came to believe that colonizing space would lead to the perfection of the
→ human species, with immortality and a carefree existence. Additionally, inspired by
→ the fiction of Jules Verne, Tsiolkovsky theorized many aspects of space travel and
→ rocket propulsion. He is considered the father of spaceflight and the first person
→ to conceive the space elevator, becoming inspired in 1895 by the newly constructed
→ Eiffel Tower in Paris.
- 2 Despite the youth's growing knowledge of physics, his father was concerned that he would
→ not be able to provide for himself financially as an adult and brought him back home
→ at the age of 19 after learning that he was overworking himself and going hungry.
→ Afterwards, Tsiolkovsky passed the teacher's exam and went to work at a school in
→ Borovsk near Moscow. He also met and married his wife Varvara Sokolova during this
→ time. Despite being stuck in Kaluga, a small town far from major learning centers,
→ Tsiolkovsky managed to make scientific discoveries on his own.
- 3 The first two decades of the 20th century were marred by personal tragedy. Tsiolkovsky's
→ son Ignaty committed suicide in 1902, and in 1908 many of his accumulated papers
→ were lost in a flood. In 1911, his daughter Lyubov was arrested for engaging in
→ revolutionary activities.
- 4 Tsiolkovsky stated that he developed the theory of rocketry only as a supplement to
→ philosophical research on the subject. He wrote more than 400 works including
→ approximately 90 published pieces on space travel and related subjects. Among his
→ works are designs for rockets with steering thrusters, multistage boosters, space
→ stations, airlocks for exiting a spaceship into the vacuum of space, and
→ closed-cycle biological systems to provide food and oxygen for space colonies.
- 5 Tsiolkovsky's first scientific study dates back to 1880-1881. He wrote a paper called
→ "Theory of Gases," in which he outlined the basis of the kinetic theory of gases,
→ but after submitting it to the Russian Physico-Chemical Society, he was informed
→ that his discoveries had already been made 25 years earlier. Undaunted, he pressed
→ ahead with his second work, "The Mechanics of the Animal Organism". It received
→ favorable feedback, and Tsiolkovsky was made a member of the Society. Tsiolkovsky's
→ main works after 1884 dealt with four major areas: the scientific rationale for the
→ all-metal balloon, streamlined airplanes and trains, hovercraft, and rockets for
→ interplanetary travel.
- 6 In 1892, he was transferred to a new teaching post in Kaluga where he continued to
→ experiment. During this period, Tsiolkovsky began working on a problem that would
→ occupy much of his time during the coming years: an attempt to build an all-metal
→ dirigible that could be expanded or shrunk in size.

- 7 Tsiolkovsky developed the first aerodynamics laboratory in Russia in his apartment. In
→ 1897, he built the first Russian wind tunnel with an open test section and developed
→ a method of experimentation using it. In 1900, with a grant from the Academy of
→ Sciences, he made a survey using models of the simplest shapes and determined the
→ drag coefficients of the sphere, flat plates, cylinders, cones, and other bodies.
→ Tsiolkovsky's work in the field of aerodynamics was a source of ideas for Russian
→ scientist Nikolay Zhukovsky, the father of modern aerodynamics and hydrodynamics.
→ Tsiolkovsky described the airflow around bodies of different geometric shapes, but
→ because the Society did not provide any financial support for this project, he was
→ forced to pay for it largely out of his own pocket.
- 8 Tsiolkovsky studied the mechanics of lighter-than-air powered flying machines. He first
→ proposed the idea of an all-metal dirigible and built a model of it. The first
→ printed work on the airship was "A Controllable Metallic Balloon" in which he gave
→ the scientific and technical rationale for the design of an airship with a metal
→ sheath. Tsiolkovsky was not supported on the airship project, and the author was
→ refused a grant to build the model. An appeal to the General Aviation Staff of the
→ Russian army also had no success. In 1892, he turned to the new and unexplored field
→ of heavier-than-air aircraft. Tsiolkovsky's idea was to build an airplane with a
→ metal frame. In the article "An Airplane or a Birdlike Flying Machine" are
→ descriptions and drawings of a monoplane, which in its appearance and aerodynamics
→ anticipated the design of aircraft that would be constructed 15 to 18 years later.
→ In an Aviation Airplane, the wings have a thick profile with a rounded front edge
→ and the fuselage is faired. But work on the airplane, as well as on the airship, did
→ not receive recognition from the official representatives of Russian science, and
→ Tsiolkovsky's further research had neither monetary nor moral support. In 1914, he
→ displayed his models of all-metal dirigibles at the Aeronautics Congress in St.
→ Petersburg but met with a lukewarm response.
- 9 Disappointed at this, Tsiolkovsky gave up on space and aeronautical problems with the
→ onset of World War I and instead turned his attention to the problem of alleviating
→ poverty. This occupied his time during the war years until the Russian Revolution in
→ 1917.
- 10 Starting in 1896, Tsiolkovsky systematically studied the theory of motion of rocket
→ apparatus. Thoughts on the use of the rocket principle in the cosmos were expressed
→ by him as early as 1883, and a rigorous theory of rocket propulsion was developed in
→ 1896. Tsiolkovsky derived the formula, which he called the "formula of aviation",
→ now known as Tsiolkovsky rocket equation, establishing the relationship between
→ change in the rocket's speed exhaust velocity of the engine initial and final mass
→ of the rocket. After writing out this equation, Tsiolkovsky recorded the date: 10
→ May 1897. In the same year, the formula for the motion of a body of variable mass
→ was published in the thesis of the Russian mathematician I V Meshchersky.
- 11 His most important work, published in May 1903, was Exploration of Outer Space by Means
→ of Rocket Devices. Tsiolkovsky calculated, using the Tsiolkovsky equation, that the
→ horizontal speed required for a minimal orbit around the Earth is 8000 meters per
→ second and that this could be achieved by means of a multistage rocket fueled by
→ liquid oxygen and liquid hydrogen. In the article Exploration of Outer Space by
→ Means of Rocket Devices it was suggested for the first time that a rocket could
→ perform space flight. In this article and its sequels he developed some ideas of
→ missiles and considered the use of liquid rocket engines.

- 12 The outward appearance of Tsiolkovsky's spacecraft design, published in 1903, was a
→ basis for modern spaceship design. The design had a hull divided into three main
→ sections. The pilot and copilot would occupy the first section, while the second and
→ third sections held the liquid oxygen and liquid hydrogen needed to fuel the
→ spacecraft.

coded_bio.txt

- 1 Yxntqptaxpd bfx gtws ns Nemjaxptdj (stb ns Xufxxpd Inxywnhy, Wdfefs Tgqfxy), ns ymj
→ Wzxxnfs Jrunwj, yt f rniiqj-hqfxx kfrnqd. Mnx kfymjw bfx f Utqnxm ktwjxyjw tk Wtrfs
→ Hfymtqnh kfny bmt wjqthfyji yt Wzxxnf; mnx Wzxxnfs Twymtitc rtymjw Rfwnf Nafstasf
→ Dzrfxmjaf bfx tk rncji Atqlf Yfyfw fsi Wzxxnfs twlns. Mnx kfymjw bfx xzhhjxxnajqd f
→ ktwjxyjw, yjfhmjw, fsi rnstw ltajwsrjsy tkknhnfq. Fy ymj flj tk 9, Ptsxyfsyns hfzlmj
→ xhfwqjy kjajw fsi qtxy mnx mjfwns. Bmjs mj bfx 13, mnx rtymjw inji. Mj bfx sty
→ firnyyji yt jqjrjsyfw xhmttqx gjhfzj tk mnx mjfwns uwtgqjr, xt mj bfx
→ xjqk-yfzlmj. Fx f wjhqzxnaj mtrj-xhmttqji hmnqi, mj ufxxji rzhm tk mnx ynrj gd
→ wjfinsl gttpx fsi gjhfrj nsywjxyji ns rfymjrfinhx fsi umdxnhx. Fx f yjjsfljw, mj
→ gjlfs yt htsyjrufy ymj utxxngnqnd tk xufhj ywfajq. Yxntqptaxpd xujsy ymwjj djfwx
→ fyyjsinsl f Rtxhtb qngfwf, bmjw Wzxxnfs htxrnxr uwtutsjsy Snptqfn Kditwta btpji.
→ Mj qfyjw hfrj yt gjqnaj ymfy htqtsnensl xufhj btzqi qjfi yt ymj ujwkjhynts tk ymj
→ mzfys xujhnjx, bny nrrtwyfnyd fsi f hfwjkjw jcnxyjshj. Fiinyntsqqd, nsxunwj gd
→ ymj knhynts tk Ozqjx Ajwsj, Yxntqptaxpd ymjtwnej rfsd fxujhyx tk xufhj ywfajq fsi
→ wthpjy uwtuzqxnts. Mj nx htsxnijwji ymj kfymjw tk xufhjkqnlmj fsi ymj knwxy ujwxts
→ yt htshjnaj ymj xufhj jqjafytw, gjhtrns nsxunwj ns 1895 gd ymj sjbqd htsxywzhyji
→ Jnkkjq Ytbjw ns Ufwnx.
- 2 Ijxuny ymj dtzym'x lwtbns pstbqjilj tk umdxnhx, mnx kfymjw bfx htshjwsji ymfy mj btzqi
→ sty gj fgqj yt uwtanij ktw mnrxjqk knsfshnfqqd fx fs fizqy fsi gwtzlmj mnr gfhp mtrj
→ fy ymj flj tk 19 fkyjw qjfwns ymfy mj bfx tajwbtpnsl mnrxjqk fsi ltensl mzsld.
→ Fkyjwbwix, Yxntqptaxpd ufxxji ymj yjfhmjw'x jcf fsi bjsy yt btp fy f xhmttq ns
→ Gtwtaxp sjfw Rtxhtb. Mj fqxt rjy fsi rfwnji mnx bnkj Afwafw Xtptqtaf izwnsl ymnx
→ ynrj. Ijxuny gjns xyzhp ns Pfqzlf, f xrfqy ytbs kfw kwtr rfotw qjfwns hjsywx,
→ Yxntqptaxpd rfsflji yt rfpj xhnjsynkn inxhtajwnjx ts mnx tbs.
- 3 Ymj knwxy ybt ijhfijx tk ymj 20ym hjsyzwd bjw rfwwji gd ujwxtsfq ywfljid. Yxntqptaxpd'x
→ xts Nlsfyd htrnyyji xznhnj ns 1902, fsi ns 1908 rfsd tk mnx fhhrzqfyji ufujwx
→ bjw qtxy ns f kqtti. Ns 1911, mnx ifzlmjw Qdzgta bfx fwwjxyji ktw jslflns ns
→ wjatqzyntsfwd fhynanyjx.
- 4 Yxntqptaxpd xyfyji ymfy mj ijajqtuji ymj ymjtwd tk wthpjywd tsqd fx f xzuuqjrjsy yt
→ umnqtxtumnhfq wjxjfw hm ts ymj xzgojhy. Mj bwtjy rtwj ymfs 400 btpx nshqzinsl
→ fuuwtcnrfyjqd 90 uzgqnxmji unjhjx ts xufhj ywfajq fsi wjqfyji xzgojhyx. Frtsl mnx
→ btpx fwj ijxnlx ktw wthpjy bny xyjwnsl ymwzxyjwx, rzqynxyflj gttxyjwx, xufhj
→ xyfntsx, fnwqthpx ktw jcnynsl f xufhxmnu nsyt ymj afhzr tk xufhj, fsi
→ hqtjxi-hdhqj gntqtlnhfq xdyjrx yt uwtanij ktti fsi tcdljs ktw xufhj htqtsnjx.
- 5 Yxntqptaxpd'x knwxy xhnjsynkn xyzid ifyix gfhp yt 1880-1881. Mj bwtjy f ufujw hfqqji
→ "Ymjtwd tk Lfxjx," ns bnmhm mj tzyqnsji ymj gfnx tk ymj pnsjynh ymjtwd tk lfxjx,
→ gzy fkyjw xzgrnyynsl ny yt ymj Wzxxnfs Umdxnh-Hmjrnhfq Xthnjyd, mj bfx nsktwrji
→ ymfy mnx inxhtajwnj mfi fqwjfid gjjs rfij 25 djfw jfwqnjw. Zsifzsyji, mj uwjxxji
→ fmjfi bny mnx xjhtsi btp, "Ymj Rjhmfsnhx tk ymj Fsnrfq Twlfsnxr". Ny wjhjnaji
→ kfatwfgqj kjjigfhp, fsi Yxntqptaxpd bfx rfij f rjrgjw tk ymj Xthnjyd. Yxntqptaxpd'x
→ rfns btpx fkyjw 1884 ijfy bny ktzw rfotw fwjfx: ymj xhnjsynkn wfyntsfq ktw ymj
→ fqq-rjyfq gfqqtts, xywjfrqnsji fnwuqfsjx fsi ywfnsx, mtajwhwky, fsi wthpjy ktw
→ nsyjuqfsjyfw ywfajq.

- 6 Ns 1892, mj bfx ywfsxkjwwji yt f sjb yjfhmnsł utxy ns Pfqzłf bmjwj mj htsynszji yt
→ jcujuwrjsy. Izwnsl ymnx ujwnti, Yxntqptaxpd gjłfs btwpnsl ts f uwtgqjr ymfy btzqi
→ thhzud rzhm tk mnx ynrj izwnsl ymj htrnsl djfwx: fs fyyjrui yt gznqi fs fqq-rjyfq
→ inwnłngqj ymfy htzqi gj jcufsiyi tw xmwzsp ns xnej.
- 7 Yxntqptaxpd ijajqtuji ymj knwxy fjwtdidsfrnhx qfgtwfytwł ns Wzxxnf ns mnx fufwyrjsy. Ns
→ 1897, mj gznqy ymj knwxy Wzxxnfs bnsi yzssjq bnył fs tujs yjxy xjhynts fsi ijajqtuji
→ f rjymti tk jcujuwrjsyfynts zxnsł ny. Ns 1900, bnył f łwfsy kwtr ymj Fhfijrd tk
→ Xhnjshjx, mj rfij f xzwajd zxnsł rtijqx tk ymj xnruijxy xmfujx fsi iyyjuwrnsji ymj
→ iwfl htjkknhjsyx tk ymj xumjwj, kqfy uqfyjx, hdqnsijwx, htsjx, fsi tymjw gtinjx.
→ Yxntqptaxpd'x btwp ns ymj knjqj tk fjwtdidsfrnhx bfx f xtzwhj tk nijfx ktw Wzxxnfs
→ xhnjsynxy Snptqfd Emzptaxpd, ymj kfymjw tk rtijws fjwtdidsfrnhx fsi mdiwtdidsfrnhx.
→ Yxntqptaxpd ijxhwngji ymj fnwkqtb fwtzsi gtinjx tk inkkjwjsy łjtrjywnh xmfujx, gzy
→ gjhfzxi ymj Xthnjyd ini sty uwtanij fsł knsfshnfq xzuutwy ktw ymnx uwtojhy, mj bfx
→ ktwhjy yt ufd ktw ny qfwłjqd tzy tk mnx tbs uthpjy.
- 8 Yxntqptaxpd xyzinji ymj rjhmfsnhx tk qnlmyjw-ymfs-fnw utbjwji kqdnsł rfhmnsjx. Mj knwxy
→ uwtutxji ymj nijf tk fs fqq-rjyfq inwnłngqj fsi gznqy f rtijq tk ny. Ymj knwxy
→ uwnsyji btwp ts ymj fnwxmnu bfx "F Htsywtqqfgqj Rjyfqqnł Gfqqtts" ns bmnłm mj łfaj
→ ymj xhnjsynknł fsi yjhmsnhfq wfyntsfqj ktw ymj ijsxłs tk fs fnwxmnu bnył f rjyfq
→ xmjfył. Yxntqptaxpd bfx sty xzuutwyji ts ymj fnwxmnu uwtojhy, fsi ymj fzymtw bfx
→ wjkzxi f łwfsy yt gznqi ymj rtijq. Fs fuujfq yt ymj Łjsjwfq Fanfynts Xyfkł tk ymj
→ Wzxxnfs fwd fqxt mfi st xzhhjxx. Ns 1892, mj yzwsji yt ymj sjb fsi zsjcuqtłji knjqj
→ tk młfanjw-ymfs-fnw fnwhwfky. Yxntqptaxpd'x nijf bfx yt gznqi fs fnwuqfsj bnył f
→ rjyfq kwfrj. Ns ymj fwynhqj "Fs Fnwuqfsj tw f Gnwinqpj Kqdnsł Rfhmnsj" fwj
→ ijxhnuyntsx fsi iwfbnslx tk f rtstuqfsj, bmnłm ns nyx fuujfwfshj fsi fjwtdidsfrnhx
→ fsynhnufyji ymj ijsxłs tk fnwhwfky ymfy btzqi gj htsxywzhyji 15 yt 18 djfwx qfyjw.
→ Ns fs Fanfynts Fnwuqfsj, ymj bnsłx mfaj f ymnłp uwtknqj bnył f wtzsiyi kwtsy jilj
→ fsi ymj kzxjqłłj nx kfłwji. Gzy btwp ts ymj fnwuqfsj, fx bjqq fx ts ymj fnwxmnu, ini
→ sty wjhjnaj wjhtłsnynsł kwtr ymj tkknłnfq wjuwjxjsyfynajx tk Wzxxnfs xhnjshj, fsi
→ Yxntqptaxpd'x kzymjw wjxjfwłł mfi sjnymjw rtsjyfwł stw rtwfq xzuutwy. Ns 1914, mj
→ inxuqfdji mnx rtijqx tk fqq-rjyfq inwnłngqjx fy ymj Fjwtsfzynhx Htsłwjxx ns Xy.
→ Ujyjwxgzł gzy rjy bnył f qzpbfwł wjxutsxj.
- 9 Inxfuutnsyji fy ymnx, Yxntqptaxpd łfaj zu ts xufhj fsi fjwtsfzynhfq uwtgqjrx bnył ymj
→ tsxjy tk Btwqi Bfw N fsi nsxyfłi yzwsji mnx fyyjsyntsy yt ymj uwtgqjr tk fqqjanfyńsl
→ utajwył. Ymnx thhzunji mnx ynrj izwnsl ymj bfw djfwx zsynq ymj Wzxxnfs Wjatqzyńts ns
→ 1917.
- 10 Xyfwynsl ns 1896, Yxntqptaxpd xdxjrfynhfqqd xyzinji ymj ymjtwł tk rtyńts tk wthpjy
→ fuufwyfyzx. Ymtzłmyx ts ymj zxj tk ymj wthpjy uwnshnuqj ns ymj htłrtłx bjwj jcuwjxxji
→ gd mnr fx jfwqd fx 1883, fsi f włłtwłzx ymjtwł tk wthpjy uwtuzqxńts bfx ijajqtuji ns
→ 1896. Yxntqptaxpd ijułnaji ymj ktwrzqf, bmnłm mj hfqqji ymj "ktwrzqf tk fanfyńts",
→ stb pstbs fx Yxntqptaxpd wthpjy jvzfynsł, jxyfgqnxmnsł ymj wjqfyńtsxmnu gjybjsł
→ hmfsłj ns ymj wthpjy'x xujji jcmfzxy ajqthnył tk ymj jslńsj nsynnfq fsi knsfq rfxx
→ tk ymj wthpjy. Fkyjw bwnynsl tzy ymnx jvzfynsł, Yxntqptaxpd wjhtłwiyi ymj ifyj: 10
→ Rfd 1897. Ns ymj xfrj djfw, ymj ktwrzqf ktw ymj rtyńts tk f gtid tk afwnfgqj rfxx
→ bfx uzgqnxmji ns ymj ymjxnł tk ymj Wzxxnfs rfymjrfynłnfs N A Rjxmłmjwxpd.
- 11 Mnx rtxy nrutwyfsy btwp, uzgqnxmji ns Rfd 1903, bfx Jcuqtwyfynsł tk Tzyjw Xufhj gd Rjfsx
→ tk Wthpjy Ijanłjx. Yxntqptaxpd hfqhzqfyji, zxnsł ymj Yxntqptaxpd jvzfynsł, ymfy ymj
→ mtwnetsyfq xujji wjvznłji ktw f rnsnrfq twgny fwtzsi ymj Jfwym nx 8000 rjywx ujw
→ xjłtsi fsi ymfy ymnx htzqi gj fhmłnaji gd rjfsx tk f rzqynxyfłł wthpjy kzjqji gd
→ qnvzłi tcdłjs fsi qnvzłi mdiwłłjs. Ns ymj fwynhqj Jcuqtwyfynsł tk Tzyjw Xufhj gd
→ Rjfsx tk Wthpjy Ijanłjx ny bfx xzłłjxyji ktw ymj knwxy ynrj ymfy f wthpjy htzqi
→ ujwktwr xufhj kqnlmy. Ns ymnx fwynhqj fsi nyx xjvzjqx mj ijajqtuji xtrj nijfx tk
→ rnxxnqjx fsi htsxnijłji ymj zxj tk qnvzłłł wthpjy jslńsjx.

12 Ymj tzybfwi fuujfwfshj tk Yxntqptaxpd'x xufhjhwfky ijxnls, uzgqnxmji ns 1903, bfx f
→ gfixnx ktw rtijws xufhjxmnu ijxnls. Ymj ijxnls mfi f mzqq inaniji nsyt ymwjj rfns
→ xjhyntsx. Ymj unqty fsi htunqty btzqi thhzud ymj knwxy xjhynts, bmnqj ymj xjhtsi fsi
→ ymnwi xjhyntsx mjqi ymj qnvzni tcdljs fsi qnvzni mdiwtljs sjjiji yt kzjq ymj
→ xufhjhwfky.