ОТЧЕТ К ЛАБОРАТОРНОЙ РАБОТЕ №1

Лабораторная работа №1

Шифр Цезаря

Вариант №13

Ф. И. О. студента: Мотынга Сергей Андреевич

Группа: ФИТ-231

Проверил: Дата:

Основные сведения

Прямое преобразование шифра Цезаря: это сдвиг позиции каждой буквы на **k** по модулю размера алфавита **m**.

Математическая формула: $E_k(x) = (x + k)(mod\ m)$

где:

- x – порядковый номер буквы в алфавите;
- k – ключ (сдвиг);
- m –мощность алфавита.

Так как в моем случае использовался международный стандарт кодирования символов *Unicode*, то формула была преобразована, однако суть формулы не поменялась.

Модифицированная формула: $E_k(x) = a + (x + k - a)(mod\ m)$

где:

- a – начало диапазона строчных/заглавных букв алфавита;
- все остальное как в базовой формуле.

Обратное преобразование шифра Цезаря: это сдвиг позиции каждой буквы зашифрованного текста на **k** позиций назад по алфавиту, вычисленный по модулю размера алфавита **m**.

Математическая формула: $D_k(y) = (y - k)(mod\ m)$

где:

- y – порядковый номер буквы в алфавите в зашифрованном виде;
- k – ключ (сдвиг);
- m – мощность алфавита.

Формула также была преобразована.

Модифицированная формула: $D_k(y) = b - (b - y + k)(mod\ m)$

где:

- b – конец диапазона строчных/заглавных букв алфавита;
- все остальное как в базовой формуле.

Таблица кодировки шифра Цезаря:

Заглавные:

Буква	Код	Буква	Код	Буква	Код	Буква	Код
А	1040	Б	1041	В	1042	Г	1043
Д	1044	Е/Ё	1045	Ж	1046	З	1047
И	1048	Й	1049	К	1050	Л	1051
М	1052	Н	1053	О	1054	П	1055
Р	1056	С	1057	Т	1058	У	1059
Ф	1060	Х	1061	Ц	1062	Ч	1063
Ш	1064	Щ	1065	Ъ	1066	Ы	1067
Ь	1068	Э	1069	Ю	1070	Я	1071

Строчные:

Буква	Код	Буква	Код	Буква	Код	Буква	Код
а	1072	б	1073	в	1074	г	1075
д	1076	е/ё	1077	ж	1078	з	1079
и	1080	й	1081	к	1082	л	1083
м	1084	н	1085	о	1086	п	1087
р	1088	с	1089	т	1090	у	1091
ф	1092	х	1093	ц	1094	ч	1095
ш	1096	щ	1097	ъ	1098	ы	1099
ь	1100	э	1101	ю	1102	я	1103

Результаты

ССЫЛКА НА РАБОЧУЮ ПРОГРАММУ: Шифр Цезаря

ШИФР-ТЕКСТ (ШТ):

ючнухчнсмсмптсйътснэыщщыючпнюьнштъънмьыунэыщбэнъгафаыяснън

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):

скажикадядяведьнедароммосквaспаленнаяпожаромфранцузуотдана

КЛЮЧ: 13

АВТОР ПРОИЗВЕДЕНИЯ (ОТ): лермонтовбородино

ЗАШИФРОВАННЫЕ ФАМИЛИЯ И НАЗВАНИЕ (ШТ):

штэщыъяыпоыэысхъы

Варианты расшифрования исходного ШТ при различных значениях ключах:

Ключ: 1

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):

эцмтфцмрлрлосрищсрмьъшшъэцомэымчсщщмлыътмьъшаьмщвяуяъюрмщм

Ключ: 2

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):

ьхлсухлпкпкнрпзшрплыщччщьхнльълцршшлкъщслыщчяылшбютющэплшл

Ключ: 3

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
ыфкртфкойоймпожчпокъшццшыфмкыщкхпччкйщшркъшцюъкчаэсэшьокчк

Ключ: 4

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
ъуйпсуйнинилонецонйщчххчъулйъшйфоццйишчпйщчхэщйцяьрьчынйцй

Ключ: 5

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
щтиортимзмзкнмдхнмишцффцщткищчиунххизчцоишцфьшихюыпыцъмихи

Ключ: 6

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
шсзнпсзлжлжймлгфмлзчхуухшсйзшцзтмффзжцхнзчхуычзфэъоъхщлзфз

Ключ: 7

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
чржморжкекеилквулкжцфттфчрижчхжслуужехфмжцфтъцжуьщнщфшкжуж

Ключ: 8

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
цпелнпейдйдзкйбткйехуссуцпзецферкттедфулехусщхетышмшучйете

Ключ: 9

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
ходкмодигигжйиасйидфтрртхождхудпйссдгуткдфтршфдсъчлчтцидсд

Ключ: 10

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
фнгйлнгзвзвеизяризгусппсфнегфтгоирргвтсйгуспчугрщцкцсхзгрг

Ключ: 11

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
умвикмвжбжбдзжюпзжвтроорумдвусвнзппвбсривтроцтвпшхйхрфжвпв

Ключ: 12

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
тлбзйлбеаеагжеэожебспннптлгбтрбмжообарпзбспнхсбочфифпуебоб

Ключ: 13

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
скажикадядяведьнедароммосквaспаленнаяпожаромфранцузуотдана

Ключ: 14

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
рйяезйягюгюбдгымдгяпнллнрйбяроякдммяюонеяпнлупямхтжтнсгямя

Ключ: 15

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
пиюджиювэвэагвългвюомккмпиаюпнюйгллюэнмдюомктоюлфсесмрвюлю

Ключ: 16

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
озэгезэбьбьявбщквбэнлййлозяэомэивккэьмлгэнлйснэкурдрлпбэкэ

Ключ: 17

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
нжьвджьаыаыюбашйбаьмкиикнжюьнльзбййьылквьмкирмьйтпгпкоаьйь

Ключ: 18

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
меыбгеыяъяъэаячиаяылйззймеэымкыжаииыъкйбылйзплыисовойняыиы

Ключ: 19

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
лдъавдъющющьяюцзяюъкижжилдьълйъеяззъщйиаъкижокъзрнбнимюъзъ

Ключ: 20

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
кгщябгщэшэшыюэхжюэщйзеезкгыщкищдюжжщшизящйзенйщжпмамзлэщжщ

Ключ: 21

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
йвшюавшьчьчъэьфеэьшиждджйвъшйзшгэеешчзжюшиждмишеолялжкьшеш

Ключ: 22

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
ибчэябчыцыщщьыудьычзеггеибщчижчвьддчцжеэчзеглзчднкюкейычдч

Ключ: 23

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
зацьюацъхъхшыътгыъцждввдзашцзецбыггцхедьцждвкжцгмйэйдиъцгц

Ключ: 24

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
жяхыэяхщфщфчъщсвъщхегббгжячхждхаъввхфдгыхегбйехвлиьигзщхвх

Ключ: 25

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
еюфъьюфшушуцщшрбщшфдваавеюцфегфящббфугвъфдваидфбкзызвжшфбф

Ключ: 26

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
дэущыэучтчтхшчпашчугбяябдэхудвуюшааутвбщугбязгуайжъжбечуау

Ключ: 27

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
гьтшъьтцсцсфчцоячцтваююагьфтгбтэчяятсбаштваюжвтяиещеадцтят

Ключ: 28

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
высчщысхрхруцхнюцхсбяээявыусвасьцююсраячсбяэебсюздшдягхсюс

Ключ: 29

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
бърцшърфпфптхфмэхфраюььюбътрбярыхээрпяюцраюьдарэжгчгювфрэр

Ключ: 30

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
ащпхчщпуоуосфульфупяэыыэащспаюпъфььпоюэхпяэыгяпьевцвэбупьп

Ключ: 31

РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
яшофцшотнтнруткыутоюьъъьяшрояэощуыыонэьфоюьъвюоыдбхбьатоыо

Код программы

```
import streamlit as st

EXCEPTIONS = '.,?!-" '

def isRussia(letter: str) -> bool:

    return 1040 <= ord(letter) <= 1103

def encryption(text: str, k: int) -> str:

    result = ''

    for i in range(len(text)):

        letter = text[i]

        if not isRussia(letter) or letter in EXCEPTIONS:

            result += letter
```

```
            continue
        elif letter.islower():
            result += chr(ord('a') + (ord(letter) + k - ord('a')) %
32)
        else:
            result += chr(ord('A') + (ord(letter) + k - ord('A')) %
32)
    return result
def decrypt_enumeration(text):
    result = {}
    for k in range(1, 32):
        result[k] = ''
        for i in range(len(text)):
            letter = text[i]
            if not isRussia(letter) or letter in EXCEPTIONS:
                result[k] += letter
                continue
            elif letter.islower():
                result[k] += chr(ord('я') - (ord('я') - ord(letter) +
k) % 32)
            else:
                result[k] += chr(ord('Я') - (ord('Я') - ord(letter) +
k) % 32)
    return result
def decrypt(text: str, k: int=None):
    result = ''
```

```
    if (k is None):
        result = decrypt_enumeration(text)
    else:
        for i in range(len(text)):
            letter = text[i]
            if not isRussia(letter) or letter in EXCEPTIONS:
                result += letter
                continue
            elif letter.islower():
                result += chr(ord('я') - (ord('я') - ord(letter) + k)
% 32)
            else:
                result += chr(ord('Я') - (ord('Я') - ord(letter) + k)
% 32)
    return result
def get_shift(action: str):
    if action == "Шифровка":
        shift = st.number_input(
            label='Шаг от 1 до 31',
            min_value=1, max_value=31, step=1,
            key="enc_shift"
        )
        return int(shift)
    else:
        use_step = st.checkbox("Выбрать шаг", value=False,
key="dec_use_step")
```

```
        if use_step:
            shift = st.number_input(
                label='Шаг от 1 до 31',
                min_value=1, max_value=31, step=1,
                key="dec_shift"
            )
            return int(shift)
        else:
            return None
def main_page():
    st.title("Шифр Цезаря")
    if "dec_variants" not in st.session_state:
        st.session_state.dec_variants = {}
    if "last_text_for_variants" not in st.session_state:
        st.session_state.last_text_for_variants = ""
    user_text = st.text_area("Введите текст:").replace('ё',
'e').replace('Ё', 'E')
    if user_text != st.session_state.last_text_for_variants:
        st.session_state.dec_variants = {}
        st.session_state.last_text_for_variants = user_text
    action = st.selectbox("Выберите действие:", ("Шифровка",
"Расшифровка"))
    shift = get_shift(action)
    if action == "Расшифровка" and shift is not None:
        if st.button("Очистить накопленные варианты",
key="clear_variants_btn"):
```

```
            st.session_state.dec_variants = {}
            st.info("Список вариантов очищен.")
    if st.button(action, key="do_action_btn"):
        file_name = None
        file_content = None
        if action == "Шифровка":
            result = encryption(user_text, shift)
            file_name = "Шифр_Цезаря_Шифровка.txt"
            file_content = (
                "Результат шифрования (Цезарь)\n"
                f"Исходный текст (ОТ):\n{user_text}\n\n"
                f"Ключ: {shift}\n\n"
                f"Зашифрованный текст (ШТ):\n{result}\n"
            )
            st.success("Текст зашифрован. Файл готов к скачиванию.")
        else:
            if shift is None:
                all_results = decrypt(user_text, None)
                file_name = "Шифр_Цезаря_Расшифровка_Все_Ключи.txt"
                lines = [
                    "Результат расшифровки (перебор ключей)",
                    f"ШИФР-ТЕКСТ (ШТ):\n{user_text}\n"
                ]
                for k in range(1, 32):
                    lines.append(f"Ключ: {k}")
```

```
                    lines.append(f"РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
{all_results[k]}")
                    lines.append("")
                file_content = "\n".join(lines)
                st.info("Расшифровка перебором выполнена. Файл готов к
скачиванию.")
            else:
                st.session_state.dec_variants[shift] =
decrypt(user_text, shift)
                file_name = "Шифр_Цезаря_Расшифровка.txt"
                lines = [
                    "Результаты расшифровки (выбранные ключи)",
                    f"ШИФР-ТЕКСТ (ШТ):\n{user_text}\n"
                ]
                for k in sorted(st.session_state.dec_variants.keys()):
                    lines.append(f"Ключ: {k}")
                    lines.append(f"РАСШИФРОВАННЫЙ ТЕКСТ (ОТ):
{st.session_state.dec_variants[k]}")
                    lines.append("")
                file_content = "\n".join(lines)
                st.success("Вариант добавлен/обновлён. Файл готов к
скачиванию.")
        if file_content and file_name:
            st.download_button(
                label="Скачать файл",
                data=file_content,
```

```
        file_name=file_name,
        mime="text/plain",
        key="download_btn"
    )
```