

Задание выполнил студент группы 2345 Романенко Кирилл

Задание 1

Одна из кодировок не разбиралась на уроке и чтобы ее раскодировать нужно немного поискать.

В этом задании вам необходимо “расшифровать” 5 “шифротекстов”:

Открыл онлайн-декодер <https://www.artlebedev.ru/decoder/advanced/>

1. WW91IGRpZCBpdCE=

Выставил кодировки в кодировщике
Base64 → UTF-8

Base64 ↕ ← Перенести текст UTF-8 ↕

WW91IGRpZCBpdCE= × You did it! 📋

2. S2VlbiBvbiBnb2luZyE=

Base64
Дописал символ = в конце

Автоматическая кодировка ↕ ← Перенести текст

S2VlbiBvbiBnb2luZyE== × Keen on going! 📋

3. VG9wIHNIY3JldCBpbmZvcm1hdGlvbiBpcyBuZWYlHlvdS4u

Base 64
Дописал два символа == в конце

Автоматическая кодировка ↕ ← Перенести текст

VG9wIHNIY3JldCBpbmZvcm1hdGlvbiBpcyBuZWYlHlvdS4u== × Top secret information is near you.. 📋

4. KRUGC5BAO5QXGIDUOJUWG23ZEBXW4ZJB

Открыл онлайн-декодер <https://dencode.com/ru/>
HEX → Base32

KRUGC5BAO5QXGIDUOJUWG23ZEBXW4ZJB

UTF-8 UTF-16 UTF-32 KOI8-R ↕ CRLF (Win) LF (UNIX/Mac) CR (Old Mac) +03:00 Europe/Moscow ↕

☒ Декодировано

Двоичная строка

Строка Hex

HTML Escape	KRUGC5BAO5QXGIDUOJUWG23ZEBXW4ZJB
Кодирование URL	KRUGC5BAO5QXGIDUOJUWG23ZEBXW4ZJB
Punycode IDN	KRUGC5BAO5QXGIDUOJUWG23ZEBXW4ZJB
Base32	That was tricky one!

5. Q29uZ3JhdHVsYXRpb25zISBZb3UndmUgJ2RIY3J5cHRIZCcgYWxslG1lc3NhZ2VzIQ==

Base 64

Нажал «Расшифровать»

Q29uZ3JhdHVsYXRpb25zISBZb3UndmUgJ2RIY3J5cHRIZCcgYWxslG1lc3NhZ2VzIQ== ✕

Congratulations! You've 'decrypted' all messages!

В результате вы получите 5 осмысленных фраз на английском языке.

Задание 2

“Расшифруйте” следующие “шифротексты”:

1. 4c6f72656d20497073756d2069732073696d706c792064756d6d792074657874206f6620746865207072696e74696e6720616e64207479706573657474696e6720696e6475737472792e

HEX

4c6f72656d20497073756d2069732073696d706c792064756d6d792074657874206f6620746865207072696e74696e6720616e64207479706573657474696e6720696e6475737472792e

UTF-8 UTF-16 UTF-32 KOI8-R CRLF (Win) LF (UNIX/Mac) CR (Old Mac) +03:00 Europe/Moscow

☒ Декодировано

Двоичная строка

Строка Hex

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

2. 436865636b206f757420746869732074616c6b20696620796f75206861766e277420646f6e6520736f2068747470733a2f2f7777772e796f75747562652e636f6d2f77617463683f763d6d4b535136446a427a3377

HEX

436865636b206f757420746869732074616c6b20696620796f75206861766e277420646f6e6520736f2068747470733a2f2f7777772e796f75747562652e636f6d2f77617463683f763d6d4b535136446a427a3377

UTF-8 UTF-16 UTF-32 KOI8-R CRLF (Win) LF (UNIX/Mac) CR (Old Mac) +03:00 Europe/Moscow

☒ Декодировано

Двоичная строка

Строка Hex

Check out this talk if you havn't done so <https://www.youtube.com/watch?v=mKSQ6DjBz3w>

В результате вы получите осмысленные фразы на английском языке.

Задание 3

Во всех последующих заданиях вы будете работать с “сырыми” байтами. Однако у большей части байт нет соответствующего печатного символа (например, 0x00 будет в лучшем случае напечатан как пробел), поэтому данные для заданий будут даваться в кодировках base64 и hex.

Декодируйте строку:

49276d206b696c6c696e6720796f757220627261696e206c696b65206120706f69736f6e6f7573206d757368726f6f6d

Результат представьте в base64. На выходе должно получиться:

SSdtlGtpbGxpbmcgeW91ciBicmFpbiBsaWtlIGEgcG9pc29ub3VzIG11c2hyb29t

Задание взято из <http://cryptopals.com/sets/1/challenges/1>

Исходная строка HEX:
49276d206b696c6c696e6720796f757220627261696e206c696b65206120706f69736f6e6f7573206d757368726f6f6d

Прогнал через онлайн-декодер

49276d206b696c6c696e6720796f757220627261696e206c696b65206120706f69736f6e6f7573206d757368726f6f6d

UTF-8 UTF-16 UTF-32 KOI8-R CRLF (Win) LF (UNIX/Mac) CR (Old Mac) +03:00 Europe/Moscow

☒ Декодировано

Двоичная строка

Строка Нех I'm killing your brain like a poisonous mushroom

HEX
I'm killing your brain like a poisonous mushroom

Закодировал в Base64

I'm killing your brain like a poisonous mushroom

UTF-8 UTF-16 UTF-32 KOI8-R CRLF (Win) LF (UNIX/Mac) CR (Old Mac) +03:00 Europe/Moscow

☒ Декодировано

☒ Закодировано

Двоичная строка	010010010010011101101101001000000110101101101001011011000110110001
Строка Нех	49276d206b696c6c696e6720796f757220627261696e206c696b65206120706f69736f6e6f7573206d757368726f6f6d
HTML Escape (Basic)	I'm killing your brain like a poisonous mushroom
HTML Escape (Fully)	I'm kiling
Кодирование URL	I%27m%20killing%20your%20brain%20like%20a%20poisonous%20mushroom
Punycode IDN	I'm killing your brain like a poisonous mushroom
Base32	JETW2IDLNFWDY2LOM4QHS33VOIQGE4TBNFXCA3DJNNSAYJA0BXWS43PNZX
Base45	7B9 ZD2QD VDZEDU1D\$FF6%EA44VKEZEDK44WEDDZCOCCS9E+ED-3E34E.OEV
Base64	SSdtlGtpbGxpbmcgeW91ciBicmFpbiBsaWtlIGEgcG9pc29ub3VzIG11c2hyb29t

Задание 4

Напишите функцию, которая принимает на вход две последовательности байт одинаковой длины и возвращает их побайтовый XOR.

Ваша функция работает правильно, если приняв на вход hex-декодированные `1c0111001f010100061a024b53535009181c` и `686974207468652062756c6c277320657965` вернет hex-декодированное значение: `746865206b696420646f6e277420706c6179`

В следующем здании мы будем искать уязвимости в нашем свежеиспеченном шифре!

```
def xor(a: str, b: str) -> str:
    if len(a) != len(b):
        return raise ValueError("Input strings not equal length")

    hex(int(a, 16) ^ int(b, 16))

def main():
    hex1 = '1c0111001f010100061a024b53535009181c'
    hex2 = '686974207468652062756c6c277320657965'
    ans = '746865206b696420646f6e277420706c6179'

    result = xor(hex1, hex2)

    print[f"correct? {ans == result}, result: {result}")

if __name__ == '__main__':
    main()
```