

Лабораторная работа №1. Шифр Цезаря

ФИО студента: Курпенов К. И.

Группа: ФИТ-222

Вариант: 26

Проверил: Белим С. Ю.

Основные сведения

Прямое преобразование шифра Цезаря:

$$y_i = E_k(x_i) = (x_i + k) \mod m$$

Обратное преобразование шифра Цезаря:

$$x_i = D_k(y_i) = (y_i - k) \mod m$$

Таблица кодировки символов: коды 0-32 соответствует буквам а-я в нижнем регистре. Буква ё игнорируется.

Результаты

Шифр-текст: реуцърушутъщфыьпйакряуэюушюояыьщцдъцьтуфтоцтбжощъйящц.

Расшифрованный текст: вчеловекедолжнобытьвсепрекрасноилицоиодеждаидушаимысли.

Ключ: 14.

Автор и произведение: чеховдядявня.

Зашифрованные фамилия и название: шжщпгеаеагбоа.

Варианты расшифровки при различных значениях ключа:

- ключ = 1: чеховдядявня
- ключ = 2: цдфнбгюгюбямю
- ...
- ключ = 31: щзчрджбжбдвпб

Код программы

```
encryptor.rs Rust

pub fn encrypt(abc: &Vec<char>, s: &String, key: usize) -> String {
    let mut encrypted_s = String::new();
    let key = key % abc.len();

    for c in s.chars() {
        let c_position = abc.iter().position(|symbol| *symbol == c);
        if c_position.is_none() {
            encrypted_s.push(c);
        } else {
            let encrypted_c_index = (c_position.unwrap() + key) % abc.len();
            encrypted_s.push(abc[encrypted_c_index]);
        }
    }

    encrypted_s
}
```

```
decryptor.rs Rust

pub fn decrypt(abc: &Vec<char>, s: &String, key: usize) -> String {
    let mut decrypted_s = String::new();
    let key = key % abc.len();

    for c in s.chars() {
        let decrypted_c_index = ((abc.len() as i32
            + (abc.iter().position(|symbol| *symbol == c).unwrap() as i32 - key as i32))
            % abc.len() as i32) as usize;

        decrypted_s.push(abc[decrypted_c_index]);
    }

    decrypted_s
}
```