

Функция для преобразования буквы в число (алфавитный порядок)

```
function letter_to_number(letter::Char)::Int
```

```
    return lowercase(letter) - 'a' + 1
```

```
end
```

Функция для преобразования числа в букву

```
function number_to_letter(number::Int)::Char
```

```
    return Char(number - 1 + Int('A'))
```

```
end
```

Основная функция шифрования гаммированием

```
function gamma_encrypt(text::String, key::String)::String
```

```
    # Преобразуем текст и ключ в числовые последовательности
    text_numbers = [letter_to_number(c) for c in text]
    key_numbers = [letter_to_number(c) for c in key]

    # Если ключ короче текста, повторяем его циклически
    key_length = length(key_numbers)
    extended_key = [key_numbers[(i % key_length) == 0 ? key_length : i % key_length]
        for i in 1:length(text_numbers)]

    # Выполняем сложение по модулю 33
    encrypted_numbers = [(text_numbers[i] + extended_key[i]) % 33 for i in
        1:length(text_numbers)]

    # Преобразуем числа обратно в буквы
    encrypted_text = join([number_to_letter(num) for num in encrypted_numbers])

    return encrypted_text
```

```
end
```

Исходные данные

text = "ПРИКАЗ"

key = "ТАММА"

Шифрование

encrypted_text = gamma_encrypt(text, key)

Вывод результата

println("Зашифрованный текст: ", encrypted_text)