```
---
title: "Лабораторная работа №1"
format:
 html:
   toc: true
   code-fold: true
   code-line-numbers: true
   code-tools: true

---

## Тема: Шифр простой замены

**Выполнила:** Исламова Сания Маратовна (студ. билет 1132249576)

---

``` {julia}
#| label: cipher-program
#| fig-cfp: "Реализация шифров Цезаря и Атбаш"
#| code-line-numbers: "true"
#| warning: false
#| eval: false

## Шифр Цезаря



function main()

    # Создаем алфавит для шифрования - все русские буквы в нижнем регистре

    alphabet = collect("абвгдеёжзийклмнопрстуфхцчшщъыьэюя")
    # Сохраняем длину алфавита для использования в модульной арифметике

    n = length(alphabet)

    # Бесконечный цикл для работы программы до команды выхода
    while true
        # Выводим меню с доступными командами
        println("Введите Ш чтобы зашифровать сообщение, Р чтобы расшифровать и В
чтобы выйти")
        print(">>> ")  # Приглашение для ввода команды

        # Читаем ввод пользователя, приводим к нижнему регистру и удаляем пробелы
        menu = lowercase(strip(readline()))

        # Проверяем команду выхода из программы
        if menu == "в"
            println("Выход из программы...")
            break  # Прерываем цикл while
        # Проверяем команду шифрования
        elseif menu == "ш"
            operation = "шифрование"  # Сохраняем тип операции для вывода
        # Проверяем команду расшифрования
        elseif menu == "р"
            operation = "расшифрование"  # Сохраняем тип операции для вывода
        # Если введена неизвестная команда
        else
            println("Неверная команда! Попробуйте снова.")
```
```

```
            continue  # Переходим к следующей итерации цикла
        end

        # Запрашиваем строку для обработки
        print("Введите строку: ")
        # Читаем строку, приводим к нижнему регистру и удаляем пробелы по краям
        message = lowercase(strip(readline()))

        # Запрашиваем ключ шифрования
        print("Введите ключ: ")
        # Обрабатываем ввод ключа с проверкой ошибок
        try
            # Парсим введенную строку в целое число
            key = parse(Int, readline())
        catch e
            # Если произошла ошибка (введено не число)
            println("Ошибка: введите целое число для ключа!")
            continue  # Переходим к следующей итерации цикла
        end

        # Если выбрано расшифрование, инвертируем ключ
        # (шифрование: сдвиг вперед, расшифрование: сдвиг назад)
        if menu == "p"
            key = -key  # Умножаем ключ на -1
        end

        # Инициализируем пустую строку для результата
        output = ""

        # Проходим по каждому символу введенной строки
        for letter in message
            # Ищем позицию текущей буквы в алфавите
            idx = findfirst(isequal(letter), alphabet)

            # Если буква найдена в алфавите (не пробел, не знак препинания и т.д.)
            if idx !== nothing
                # Вычисляем новую позицию с учетом ключа и длины алфавита
                # mod обеспечивает циклический сдвиг (если вышли за границы - начинаем сначала)
                # -1 и +1 нужны из-за 1-индексации в Julia (индексы начинаются с 1, а не с 0)
                new_idx = mod(idx + key - 1, n) + 1
                # Добавляем зашифрованную/расшифрованную букву к результату
                output *= string(alphabet[new_idx])
            else
                # Если символ не из алфавита (пробел, запятая и т.д.), добавляем его как есть
                output *= string(letter)
            end
        end

        # Выводим результат с указанием типа операции
        println("Результат $operation: $output")
        # Разделительная линия для визуального отделения результатов
        println("-" ^ 50)
    end
end

# Запуск программы - вызываем основную функцию
main()
```

```
## Шифр Атбаш


function main()

    # Создаем русский алфавит для шифрования Атбаш
    # Атбаш - шифр подстановки, где первая буква заменяется на последнюю,
    # вторая - на предпоследнюю, и т.д.

    alphabet = collect("абвгдеёжзийклмнопрстуфхцчшщъыьэюя")
    # Сохраняем длину алфавита

    n = length(alphabet)

    # Бесконечный цикл для работы программы до команды выхода
    while true
        # Выводим меню с доступными командами
        println("Введите Ш чтобы зашифровать сообщение, Р чтобы расшифровать и В
чтобы выйти")
        print(">>> ")  # Приглашение для ввода команды

        # Читаем ввод пользователя, приводим к нижнему регистру и удаляем пробелы
        menu = lowercase(strip(readline()))

        # Проверяем команду выхода из программы
        if menu == "в"
            println("Выход из программы...")
            break  # Прерываем цикл while
        # Проверяем команду шифрования
        elseif menu == "ш"
            operation = "шифрование"  # Сохраняем тип операции для вывода
        # Проверяем команду расшифрования
        elseif menu == "р"
            operation = "расшифрование"  # Сохраняем тип операции для вывода
        # Если введена неизвестная команда
        else
            println("Неверная команда! Попробуйте снова.")
            continue  # Переходим к следующей итерации цикла
        end

        # Запрашиваем строку для обработки
        print("Введите строку: ")
        # Читаем строку, приводим к нижнему регистру и удаляем пробелы по краям
        message = lowercase(strip(readline()))

        # Инициализируем пустую строку для результата
        output = ""

        # Проходим по каждому символу введенной строки
        for letter in message
            # Ищем позицию текущей буквы в алфавите
            idx = findfirst(isequal(letter), alphabet)

            # Если буква найдена в алфавите (не пробел, не знак препинания и т.д.)
            if idx !== nothing
                # В шифре Атбаш преобразование симметрично:
```

```
                # Шифрование и расшифрование выполняются одинаково -
                # буква с индексом i заменяется на букву с индексом (n - i + 1)
                # Например:
                # 'а' (1 позиция) -> 'я' (33 позиция при n=33)
                # 'б' (2 позиция) -> 'ю' (32 позиция)
                # и т.д.
                new_idx = n - idx + 1
                # Добавляем преобразованную букву к результату
                output *= string(alphabet[new_idx])
            else
                # Если символ не из алфавита (пробел, запятая и т.д.), добавляем его как есть
                output *= string(letter)
            end
        end

        # Выводим результат с указанием типа операции
        println("Результат $operation: $output")
        # Разделительная линия для визуального отделения результатов
        println("-" ^ 50)
    end
end

# Запуск программы - вызываем основную функцию
main()
```