

## Часть 1.

Создайте функции для шифрования и дешифрования сообщений с помощью кода Цезаря.

Код Цезаря сдвигает каждую букву сообщения на заданное количество позиций в алфавите. Например, для сдвига на 3:

- "A" станет "D"
- "B" станет "E"
- и так далее.

Если сдвиг выходит за пределы алфавита, он должен «заикливаться» в начало алфавита. Например, "Z" при сдвиге на 1 станет "A". Учитывайте также регистр букв: заглавные остаются заглавными, строчные остаются строчными.

Символы, которые не являются буквами латинского алфавита, остаются неизменными.

1. [0.5 балла] Реализуйте функцию `caesarEncrypt`, которая принимает строку и целое число `shift` и возвращает зашифрованную строку.  
Пример: `caesarEncrypt "Hello, World!" 3` должно вернуть `"Khoor, Zruog!"`.
2. [0.5 балла] Реализуйте функцию `caesarDecrypt`, которая принимает зашифрованную строку и `shift`, используемый для шифрования, и возвращает исходное сообщение.  
Пример: `caesarDecrypt "Khoor, Zruog!" 3` должно вернуть `"Hello, World!"`.
3. [0.5 балла] Реализуйте функцию `caesarCrack` для взлома кода Цезаря. Функция должна принимать зашифрованную строку и возвращать список всех возможных расшифровок, используя сдвиг от 1 до 25. Например:  
`caesarCrack "Khoor, Zruog!"` должно вернуть список из всех возможных расшифровок, включая `"Hello, World!"`.

## Часть 2

1. [0.5 балла] Напишите функцию, которая проверяет, являются ли все символы в строке уникальными. Например, `"abcde"` состоит из уникальных символов, а `"abca"` — нет.
2. [0.5 балла] Напишите функцию, которая удаляет из строки все гласные буквы. Например, для строки `"Hello World!"` результатом будет `"Hll Wrld!"`. Учитывайте как заглавные, так и строчные гласные.
3. [0.5 балла] Напишите функцию, которая проверяет, являются ли две строки анаграммами, игнорируя пробелы и регистр. Например, строки `"Listen"` и `"Silent"` являются анаграммами, а `"Hello"` и `"World"` — нет.
4. [0.5 балла] Напишите функцию, которая находит самый длинный общий префикс в списке строк. Например, для `["flower", "flow", "flight"]` результат будет `"fl"`. Если общего префикса нет, вернуть пустую строку.
5. [0.5 балла] Напишите функцию, которая возвращает список анаграмм слова. Функция принимает слово и список всех возможных слов, и возвращает те слова из этого списка, которые являются анаграммой исходного слова (слово не является анаграммой самого себя). Игнорируйте регистр при сравнении, возвращайте слова в том же регистре, в котором они находятся в списке всех возможных слов.

```
> anagramsFor "Stone" ["stone", "tones", "banana", "tons", "notes", "Seton"]  
["tones", "notes", "Seton"]
```

7. [0.5 балла] Реализуйте функцию `mySplit :: Char -> String -> [String]`, которая принимает символ-разделитель и строку, и возвращает список подстрок, полученных путем разбиения исходной строки по этому разделителю.

Например, если разделитель — пробел, строка `"Mars Base Alpha"` будет разбита на `["Mars", "Base", "Alpha"]`. При этом:

- Все пустые подстроки, которые могли бы образоваться из-за нескольких разделителей подряд, должны быть удалены.
- Начальные и конечные разделители также должны игнорироваться.