

## Задачи:

1. Да се напише функция **len()**, която приема символен низ и намира дължината му.  
Примерен вход и изход: *len("Gosho42") -> 7*
2. Да се напише функция, която приема два символни низа и връща низа, и записва във втория само малките букви на първия.  
Примерен вход и изход: *smallLetters("Gosho42", char\* result) -> result = "osho"*
3. Да се напише функция **replaceSymbol()**, която приема низ и два символа и замества всяко срещане на първия символ в дадения низ с втория.  
Примерен вход и изход: *replaceSymbol("aabcde", 'a', 'f') -> "ffbcde"*
4. Да се напише функция **removeSymbol()**, която приема символ и низ и премахва всички срещания на символа.  
Примерен вход и изход: *removeSymbol("starling", 'l') -> "starting"*
5. Да се напише функция **isPalindrome()**, която приема като аргумент символен низ и определя дали той е палиндром.  
Примерен вход и изход: *isPalindrome("abcdefhk") -> false,*  
*isPalindrome("abababa") -> true*
6. Да се напише функция **getFirstUniqueChar()**, която приема символен низ и връща първия елемент, който не се повтаря. Ако не съществува, функцията извежда 0.  
Примерен вход и изход: *getFirstUniqueChar("pepsi") -> 'e',*  
*getFirstUniqueChar("Pepsi") -> 'P'*
7. Да се напише функция **reverseString()**, която приема като параметър символен низ и го обръща.  
Примерен вход и изход: *reverseString("desserts") -> ("stressed")*
8. Да се напише функция **concat()**, която приема 3 символни низа и записва в третия конкатенацията на първите два низа, т.е първия, долепен до втория.  
Примерен вход и изход: *concat("abc", "def", char\* result) -> result = "abcdef"*

