

# Задачи:

1. Да се направи имплементация на класа `'String' MyString`  
Този клас трябва да съдържа следните методи:

- `at(index)` - Връща елемент от низа на определен `index`
- `front()` - Връща първия елемент на низа
- `back()` - Връща последния елемент на низа
- `str()` - Връща самия низ, който се съдържа в `String`
- `length()` - Връща дължината на низа
- `capacity()` - Връща количеството заделена памет за низа
- `empty()` - Връща дали стринга е празен
- `append(<подходящ/и параметър/и>)` - Долепя нов символ, низ или друг стринг
- `clear()` - Изчиства съдържанието на обекта, без да променя капацитета
- `equals(<подходящ/и параметър/и>)` - Проверява дали два низа са равни
- `find(symbol)` - Намира първото срещане на символ и връща индекса. Ако не намери, то да върне `-1`

За домашна работа

- `shrinkToFit()` - Прави капацитета равен на размера `+ 1` (за `'\0'`)
- `swap(string)` - Разменя стойностите на два обекта
- `find(string)` - Намира първото срещане на низ и връща индекса на първата буква, ако не намери - връща `-1`
- `contains(<подходящ/и параметър/и>)` - Проверява дали низ или символ се съдържа в обекта
- `copy(source)` – копира съдържанието в текущия обект
- `cmp(MyString)` – сравнява лексикографски текущия обект с подадения параметър. Ако първия е по- голям, връща `1`, ако вторият е по- голям лексикографски, връща `-1`, а ако са равни връща `0`
- `strInstr(Mystring)` – връща указател към първото срещане на текущия обект в `MyString`, или `nullptr`, ако вторият стринг не е част от първия.

Условия за решението на задачата:

- Когато създаваме обект, трябва да гарантираме, че е във валидно състояние.
- Валидация на подадените аргументи на методите
- Когато ни е нужна повече памет за долепяне на елементи, нека размера се умножава по 2. (тоест ако капацитета ни е 8, а искаме да добавим 8ми елемент, то новият капацитет ще бъде 16)

2. Да се реализира клас пазарна количка, който съдържа списък от продукти в количката. Всеки продукт съдържа име, цена за брой и количество. Да се реализират нужните селектори и мутатори. За продукта да се реализират методи за намиране на общата цена и за извеждане на подробна информация за продукта. За количката да се реализират методи за добавяне и премахване на продукт, за намиране на общата цена на продуктите в количката и за извеждане на подробна информация за количката и продуктите в нея. Да се демонстрира коректността на реализацията чрез примерна програма.
3. Да се реализира клас задача, който се описва с име, състояние (незавършена, в прогрес, завършена) и описание. Да се реализират нужните селектори и мутатори и да се реализират методи за промяна на състоянието и за извеждане на подробна информация за задачата. Да се реализира клас описващ списък от задачи. Да се реализират методи за намиране на всички задачи с дадено състояние, за добавяне, премахване и промяна на задача и за извеждане на всички задачи. Да се демонстрира коректността на реализацията чрез примерна програма.