

# ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА САМОПОДГОТОВКА

ПО

## Увод в програмирането *функции 2*

*email: kalin@fmi.uni-sofia.bg*

9 ноември 2015 г.

1. Задача 3. 23.

Да се дефинира функция, която сравнява лексикографски два символни низа. Функцията да връща 0, ако символните низове съвпадат, отрицателно цяло число, ако първият низ е по-малък от втория и положително цяло число, ако първият низ е по-голям от втория.

2. Задача 3.26.

Да се дефинира функция, която премахва повтарящите се интервали в даден символен низ.

3. Задача 3.3.

Да се напише функция, която намира и извежда сумата от положителните и произведението на отрицателните елементи на редицата от реални числа  $a_0, a_1, \dots, a_{n-1}$  ( $1 \leq n \leq 30$ ), подадена като параметър на функцията.

4. Задача 3.15.

Да се напише функция, която намира и връща скаларното произведение на реалните вектори  $a_0, a_1, \dots, a_{n-1}$  и  $b_0, b_1, \dots, b_{n-1}$  ( $1 \leq n \leq 30$ ), подадени като параметри на функцията.

5. Задача 3.1.29.

Да се напише булева функция, която определя дали векторите  $a_0, a_1, \dots, a_{n-1}$  и  $b_0, b_1, \dots, b_{n-1}$  ( $1 \leq n \leq 30$ ), подадени като параметър, са линейно зависими.

6. Задача 4.14.

Да се напише функция, която получава като параметър едномерен масив от реални числа и намира и извежда повтарящите се елементи в него. Всеки повтарящ се елемент да се изведе само веднъж.

7. Задача 4.15.

Да се напише функция, която обръща елементите на подаден едномерен масив.

8. Да се дефинира функция, която получава като масив коефициентите на реалния полином  $a_0x^n + a_1x^{n-1} + \dots + a_n$  и стойността на  $x$ . Функцията да връща стойността на полинома за тези  $a_i$  и  $x$ .

9. Задача 3.33.

Да се дефинират следните функции за работа със символни низове:

- (а) `void ToLower(char* text)`, която заменя всички главни букви от низа със съответните малки;
- (б) `void ToUpper(char* text)`, която заменя всички малки букви от низа със съответните главни;
- (в) `int StringCompareNotSensitive(const char* first, const char* second)`, която сравнява лексикографски два символни низа, като не различава малки и главни букви. Функцията да връща 0, ако символните низове съвпадат без да се различават малки и главни букви, отрицателно цяло число, ако първият низ е по-малък от втория и положително цяло число, ако първият низ е по-голям от втория;
- (г) `void TrimLeft(char* text, const char* tokens)`, която премахва от низа `text` всички символи – разделители пред първия символ, който не е разделител. Разделителите се задават чрез низа `tokens`;
- (д) `void TrimRight(char* text, const char* tokens)`, която премахва от низа `text` всички символи – разделители след последния символ, който не е разделител. Разделителите се задават чрез низа `tokens`;

- (е) `void Trim(char* text, const char* tokens)`, която премахва от низа `text` всички символи – разделители пред първия символ, който не е разделител и след последния символ, който не е разделител. Разделителите се задават чрез низа `tokens`;
- (ж) `void RemoveToken(char* text, const char* tokens)`, която премахва всички символи – разделители от низа `text`. Разделителите се задават чрез низа `tokens`;

Някои от задачите са от сборника *Магдалина Тодорова, Петър Армянов, Дафина Петкова, Калин Николов, "Сборник от задачи по програмиране на C++. Първа част. Увод в програмирането"*. За тези задачи е запазена номерацията в сборника.