

## Увод в програмирането

спец. Информатика

Задачи за задължителна самоподготовка

07.01.2014

**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.

**Задача 3.26.** Да се дефинира функция, която изтрива повтарящите се интервали в даден символен низ.

**Задача 3.33.** Да се дефинират следните функции за работа със символни низове:

а) `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;

б) `void TrimLeft(char* text, const char* tokens)`, която изтрива от низа `text` всички символи – разделители пред първия символ, който не е разделител. Разделителите се задават чрез низа `tokens`;

в) `void TrimRight(char* text, const char* tokens)`, която изтрива от низа `text` всички символи – разделители след последния символ, който не е разделител. Разделителите се задават чрез низа `tokens`;

г) `void Trim(char* text, const char* tokens)`, която изтрива 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`;

д) `void RemoveToken(char* text, const char* tokens)`, която изтрива всички символи – разделители от низа `text`. Разделителите се задават чрез низа `tokens`;

е) `void ToUpper(char* text)`, която заменя всички малки букви от низа със съответните главни;

ж) `void ToLower(char* text)`, която заменя всички главни букви от низа със съответните малки.

**Задача 3.39.** Да се напише програма, която намира и извежда на екрана всички  $n$ -мерни вектори с компоненти 0 и 1 ( $1 \leq n \leq 20$ ). Например при  $n = 3$ , всички 3-мерни вектори с компоненти 0 и 1 са: (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1).

**Задача 5.48.** Да се напише програма, която решава символния ребус `SEND + MORE = MONEY`. На равните символи в сумата съответства една и съща цифра.

**Задача 5.43.** Да се напише рекурсивна програма, която намира и извежда на екрана всички  $n$ -мерни вектори с компоненти целите числа 0, 1, ...,  $k-1$ , където  $n$  и  $k$  са дадени положителни цели числа.

**Задача 5.44.** Дадени са естествените числа  $n$  и  $k$ ,  $0 < n \leq k$ , както и редицата от  $k$  на брой различни числа  $v_0, v_1, \dots, v_{k-1}$ . Да се напише рекурсивна програма, която намира и извежда на екрана всички  $n$ -мерни вектори с компоненти  $v_0, v_1, \dots, v_{k-1}$ , като всеки от елементите може да участва не повече от веднъж.

**Задача 5.52.** Дадена е мрежа от  $m \times n$  квадратчета ( $1 \leq m \leq 10$  и  $1 \leq n \leq 20$ ), като за всяко квадратче е определен цвят - бял или черен. Път се нарича редица от съседни във вертикално или хоризонтално направление квадратчета с един и същ цвят. Област се нарича множество от квадратчета с един и същ цвят между всеки две, от

които има път и това множество е максималното по включване с това свойство. Дадено е квадратче. Да се определи:

а) броят на квадратчетата от областта, в която се съдържа даденото квадратче;

б) броят на областите с цвят, съвпадащ с цвета на даденото квадратче, както и броят на квадратчетата на всяка от тези области;

в) броят на областите с цвят, различен от цвета на даденото квадратче, както и броят на квадратчетата на всяка от тези области.

**Задача 5.51.** Да се напише програма, която намира най-краткия път от горния ляв ъгъл до долния десен ъгъл на лабиринта от предишната задача, в случай че път съществува.

**Задача 5.55.** Да се напише програма, която въвежда от клавиатурата без грешка булев израз от вида

$\langle \text{булев\_израз} \rangle ::= t \mid f \mid \langle \text{операция} \rangle (\langle \text{операнди} \rangle) \langle \text{операция} \rangle ::= n \mid a \mid o$   
 $\langle \text{операнди} \rangle ::= \langle \text{операнд} \rangle \mid \langle \text{операнд} \rangle, \langle \text{операнди} \rangle \langle \text{операнд} \rangle ::= \langle \text{булев\_израз} \rangle,$

където  $t$  и  $f$  означават истина и лъжа съответно,  $n$  има само един операнд, а  $a$  и  $o$  могат да имат произволен брой операнди и означават съответно логическо отрицание, конюнкция и дизюнкция. Програмата да намира и извежда стойността на булевия израз.

**Задача 5.56.** (Ханойски кули) С тази задача е свързана една легенда. Във великия храм на Брама под купола, разположен точно в центъра на света, лежат 64 златни диска, които са поставени един върху друг нанизани на 3 елмазени сърцевини. Жреците на Брама пренасят дисковете един по един от едната сърцевина на другата строго

съблюдавайки закона на Брама– голям диск не трябва да се поставя върху по-малък. В началото всички 64 диска се намирили на едната сърцевина, подредени по големина, като в основата се намирал най-големият диск, образувайки кулата на Брама. Когато всички дискове бъдат пренесени на другата сърцевина, ще настъпи краят на света.

Да се напише програма за пренасяне на  $k$  диска от едната сърцевина на другата като се използва една помощна сърцевина.