



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

Контролно 1

курс Увод в програмирането
за специалност Компютърни науки
зимен семестър 2021/2022 г.

Задача 1. (1,25 т.) Конкатенация на две естествени числа **A** и **B** наричаме число **A ◦ B**, което се получава като към цифрите на **A** се долепят цифрите на **B**.

Пример:

$$123 \circ 431 = 123431$$

$$123 \circ 0 = 1230$$

Да се напише програма, която въвежда две естествени числа **N** и **K** и извежда дали съществуват естествени числа **A** и **B**, такива че **N = A ◦ K ◦ B**.

*За реализацията на задача може да се използва и функция с два параметъра – **N** и **K**. В решението на задачата не се допуска използването на масиви!*

Примери:

Вход	Изход	Обяснение
N = 1234 K = 23	True	N = 1 ◦ K ◦ 4
N = 1234 K = 4	False	
N = 1234 K = 2	True	N = 1 ◦ K ◦ 34

Задача 2. (1,25 т.) Едно число ще наричаме „съседно“, ако всеки две негови съседни цифри са различни.

Пример: 2572 е съседно, но 3119 не е.

Да се напише програма, която намира разликата на най-голямото и най-малкото „съседно“ число в интервала **[a; b]**, където **a** и **b** са въведени от клавиатурата естествени числа. Ако в интервала няма съседни числа, да се изведе подходящо съобщение. Позволено е използването на масиви и функции.

Примери:

Вход	Изход	Обяснение
a = 9 b = 100	89	89 = 98 - 9

a = 1099 b = 1222	18	18 = 1219 - 1201
a = 110 b = 115	No adjacent numbers!	първите две цифри са равни

Задача 3. (1,5 т.) “Подредица” на дадена редица наричаме редица, която се получава след евентуалното изтриване на нула или повече елементи в началото и края на първоначалната редица.

Пример: редиците **abc**, **cd**, **de** са три подредици на редицата **abcde**.

“Ротация” на дадена редица се нарича такава подредба на елементите на редицата, която се получава като няколко елемента от края на редицата се преместят в същия ред в началото на редицата.

Пример: всички възможни ротации на редицата **abcde** са **eabcd**, **deabc**, **cdeab** и **bcdea**.

Да се напише програма, която въвежда от стандартния вход две положителни цели числа `size1` и `size2` и две редици от латински букви `firstArr` и `secondArr`, съответно с дължини `size1` и `size2`. Програмата да определи дали някоя ротация на `secondArr` е подредица на `firstArr`, и ако това е така да изведе ротацията и да изведе колко символа от началото и края на `firstArr` трябва да се изтрият, за да се получи подредицата `secondArr`. Ако такава ротация няма, да се изведе подходящо съобщение.

Ограничения:

$0 < \text{size2} \leq \text{size1} < 1024$

Примери:

Вход	Изход
size1: 5 size2: 3 firstArr: 'h', 'e', 'l', 'l', 'o' secondArr: 'l', 'o', 'l'	Rotation "llo" found after deleting 2 characters from the start and 0 characters from the end
size1: 7 size2: 4 firstArr: 'g', 'o', 'o', 'd', 'b', 'y', 'e' secondArr: 'b', 'o', 'd', 'y'	No rotation found!

Бонус: (1 т.) ако програмата не намери нито една ротация, която да е подредица, да определи дали от `firstArr` може да се получи някоя ротация на `secondArr` с изтриване на символи на произволни места. Ако това е възможно, да се изведат позициите в `firstArr`, на които трябва да се изтрият символите и коя ротация се получава.

Примери:

Вход	Исход
size1: 6 size2: 3 firstArr: 'a', 'x', 'z', 't', 'c', 'y' secondArr: 'c', 'a', 't'	Rotation "atc" found after deleting characters: 1 2 5
size1: 9 size2: 4 firstArr: 'c', 'm', 'd', 'a', 'i', 'h', 'n', 'b', 'a' secondArr: 'a', 'd', 'h', 'p',	No rotation found!