

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2021-2022 г.

Задачи за домашно № 4

1. Напишете функция **char** letterCasePermutation(char* str)**, която приема низ и връща масив от низове с всички възможни пермутации, където буквите са трансформирани да бъдат малки или големи. Резултатът да бъде сортиран лексикографски. В главната функция **main** извикайте **letterCasePermutation** и отпечатайте резултата.

Примери:

Вход	Изход
a1b2	a1b2 a1B2 A1b2 A1B2
3z4	3z4 3Z4
12345	12345

2. Да се напише функция, **int replace(char *text, const char *const str1, const char *const str2)**, която заменя всички срещания на **str1** в **text** със **str2**. Като резултат **replace** да върне броя на извършените промени. Да се напише функция **main**, в която програмата да прочете от стандартния вход низовете **text**, **str1** и **str2**, да изпълни функцията **replace** върху тях и да изведе на стандартния изход броя на извършените промени, както и низа **text** след промените.

За ваше улеснение:**str1** и **str2** няма да надвишават дължина **1KB****text** в нито един момент няма да надвишава дължина **4KB**За пълнота на функцията. По желание:Ако **text** или **str1** или **str2** е невалиден указател, функцията да върне резултат -2.Ако **text** или **str1** сочи към празен низ, функцията да върне резултат -3.

Примери:

Вход	Изход
ахс х b	1 abc
My name is who? who? Chica-chica Slim Shady!	1 My name is Chica-chica Slim Shady

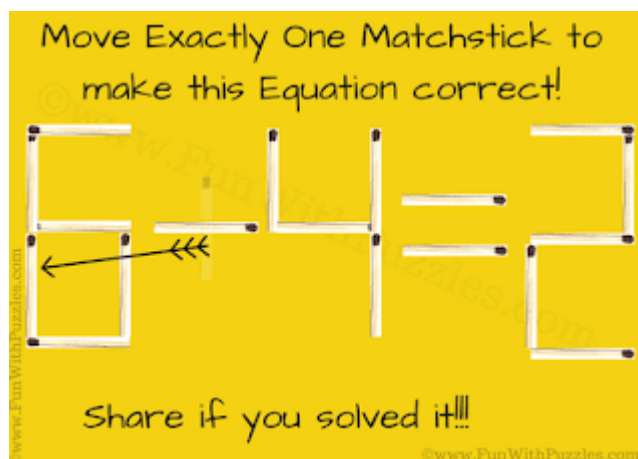
aabbaaaa	3
aa	bbbbbbbb
bb	

3. Да се напише програма, която приема символен низ с 5 символа, който представлява уравнение, което не е вярно. Нека числата се интерпретират като наредени клечки кибрит. С помощта на една операция по разместване на клечки, да се изведе правилното уравнение или "No" ако не е възможна такава трансформация. **Ако съществуват няколко възможни решения, да се изведат всичките решения, всяко на нов ред, сортирани във възходящ ред.**

Примери:

Вход	Изход
5+4=2	6-4=2
9+3=5	3+3=6 8-3=5 9-3=6
8+8=8	No

Пояснение:



Представяне на числата:



4. Напишете програма, която иска от потребителя число $N \in [3 ; 1000]$, което индикира колко числа ще бъдат въведени, след това се въвежда N пъти число $n \in \mathbb{Z}$.

Програмата да извежда на конзолата дали дадената редица от числа е hacksaw.

Hacksaw редица ще наричаме редица, която отговаря на следното условие:

$x_0 < x_1 > x_2 < x_3 > x_4 < x_5 \dots x_n$ или $x_0 > x_1 < x_2 > x_3 < x_4 > x_5 \dots x_n$.

Задачата да се реши с рекурсия.

Примери:

Вход	Изход
3 1 2 3	no
5 7 6 5 4 3	no
4 3 4 2 5	yes
6 10 2 9 3 8 7	yes
0	-1
2 2 3	-1

5. В линейното пространство \mathbb{R}^4 са дадени две числа m и n , както и векторите a_1, a_2, \dots, a_n .

Нека $U = \{a_1, a_2, \dots, a_n\}$ е подпространство на \mathbb{R}^4 ;
 $\{a_1, a_2, a_3, a_n\} = \{x | x = \sum \lambda_i a_i\}$ и нека W е
 подпространство на \mathbb{R}^4 зададено като
 пространството от решения на хомогенната система:

$$b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + b_{14}x_4 = 0$$

$$b_{21}x_1 + b_{22}x_2 + b_{23}x_3 + b_{24}x_4 = 0$$

...

$$b_{m1}x_1 + b_{m2}x_2 + b_{m3}x_3 + b_{m4}x_4 = 0$$

Да се напише функция F , която приема като параметри числата m и n ,
 векторите a_1, a_2, \dots, a_n и коефициентите на b_{ij} на хомогенната система.
 Да се напише функция main, в която са декларирани локално числата m и
 n , както и матриците $(a_{ij})_{n \times 4}$ и $(b_{ij})_{m \times 4}$, където $(a_{ij})_{n \times 4}$ е матрицата от
 координатите на векторите a_1, a_2, \dots, a_n .

Програмата да изведе на екрана на конзолата базис на $U + W$.

Да не се извежда нищо в конзолата от функцията F .

[Функцията трябва да симулира метода на Карл Фридрих Гаус за
 решаване на система линейни уравнения.](#)

Примерни данни:

Вход (примерен)	Изход (примерен)
$m = 2, n = 4$ $a_1 = (2, 8, -3, 14)$ $a_2 = (-1, 2, 3, 5)$ $a_3 = (-1, 14, 6, 29)$ $a_4 = (0, 12, 3, 24)$ $b_1 = (0, 1, 1, 0)$ $b_2 = (10, 7, 0, -8)$	$(-1, 2, 3, 5)$ $(0, 4, 1, 8)$ $(0, 0, 10, 9)$

Пояснения:

1. Всички задачи носят по 2 точки.
2. Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно от лектора и при установено плагиатство ще бъдат анулирани.
3. Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата.
4. За решаване на задачите **не се допуска** използването на *string*, *cstring* и *всякакъв STL*.
5. Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC
6. Всяка задача от домашното трябва да бъде решена в точно един, отделен файл. Името на файла трябва да бъде в следния формат:
fnXXXXXXXXXX_d4_N_CC.cpp, където:
 - XXXXXXXXXXXX е вашият факултетен номер
 - N е номерът на задачата
 - CC указва кой компилатор сте използвали. Стойността му може да бъде "gcc" за GCC или "vc" за Visual C++.
7. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:
UP_21-22_fnXXXXXXXXXX_d4.zip, където XXXXXXXXXXXX е вашият факултетен номер
8. Файловете с решенията, които предавате трябва да са оформени съгласно добрите практики за оформяне на кода, за които се говори по време на лекции и упражнения. Ще се отнемат точки за неинформативни имена на променливи, неизползване на подходящи константи и т.н.
9. Всички предадени програми трябва да следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата. При некоректни входни данни програмата да извежда на екрана -1.
10. Файловете с решенията може да съдържат само стандартните символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
11. Първото нещо във всеки от файловете, които предавате, трябва да бъде коментарен блок, който носи информация за съдържанието на файла. Този коментар трябва да изглежда точно така, както е показано по-долу, като в него попълните информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

/ **

*

```
* Solution to homework assignment 4
* Introduction to programming course
* Faculty of Mathematics and Informatics of Sofia
University
* Winter semester 2021/2022
*
* @author <вашето име>
* @idnumber <вашият факултетен номер>
* @task <номер на задача>
* @compiler <използван компилатор - GCC или VC>
*
*/
```

Например един попълнен блок за студент с име Иван Иванов, ф.н. 12345, който предава задача 4, компилирана с GCC, трябва да изглежда така:

```
/**
 *
 * Solution to homework assignment 4
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia
University
* Winter semester 2021/2022
*
* @author Ivan Ivanov
* @idnumber 12345
* @task 4
* @compiler GCC
*
*/
```

12. Предадени домашни, които не отговарят на условията от точки 4-11 ще бъдат оценени с 0 точки.