

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2020-2021 г.

Задачи за домашно № 3

1. Даден символен низ се нарича правилен, ако всички негови символи се срещат равен брой пъти в него или е достатъчно да премахнем точно един елемент, така че всички символи в новополучения низ се срещат равен брой пъти в него. Да се напише функция $F()$ ¹ на C++, която по зададен като параметър символен низ, проверява дали той е правилен. Да се напише функция `main()`, която изпълнява $F()$ върху въведен от потребителя символен низ (с максимален размер от 150 символа) и извежда в конзолата резултата от изпълнението ѝ.

Примери:

Вход	Изход
	1
	0
	1

2. Да се напише функция $F()$ ¹ на C++, която приема като параметри 3 символни низа, (всеки от тях с максимален размер 150 символа), които са съставени само от малки букви от латинската азбука и връща като резултат дали 3-тият низ може да се получи като разместим буквите в първите два низа и после ги конкатенираме (редът на конкатенция е без значение)². Да се напише функция `main()`, която изпълнява $F()$ върху 3 въведени от потребителя символни низа да се изведе резултата от изпълнението в конзолата.

¹ Сами определете подходящо име на функцията.

² Третият низ трябва да се получи точно от първите два

Примери:

Вход	Изход
	1
	0
	1
1 1	-1

3. Да се напише функция $F()$ ¹ на C++, която получава като параметри цяло число $1 < N < 1000$ и квадратна матрица с размер $N \times N$ съставена от реални числа в интервала $[0, 100]$, и връща като резултат булева стойност дали матрицата представлява магически квадрат. Да се напише функция `main()`, която изпълнява $F()$ върху въведени от потребителя размер и матрица и да изведе в конзолата резултата от изпълнението.

Пояснения:

- Магически квадрат означава, че сборът на числата във всички редове, колони и по двата диагонала, на матрицата е един и същ.
- Внимавайте с директното сравнение на реални числа.

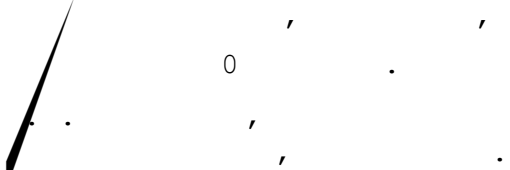

Примери:

Вход	Изход
3 8 1 6 3 5 7 4 9 2	
2 1 2.5 3 4.8	

4. Даден е символен низ, който съдържа k , $k < 100$ на брой латински букви. От него може да се задраскват 0, 1, 2, ... до $k-1$ произволни букви (конкретни елементи на низа). Да се състави функция F^1 на C++, която приема като параметър символен низ и връща

като резултат броя на различните символни низове, които могат да се получат чрез така описаното задраскване. Да се напише функция `main()`, в която пита потребителя да въведе символен низ, изпълнява `F()` върху него и да се извежда резултата в конзолата.

Примери:

Вход	Изход	Пояснение
	3	
	10	

5. Да се напише програма на C++, която разчита следния шифър:
 - От файл с име "message.txt" се прочитат всички негови редове.
 - При неуспех с отварянето или прочитането на файла, програмата да изведе в конзолата резултат -2.
 - Всеки ред съдържа не повече от 150 символа, които може да бъдат само малки и/или големи букви от латинската азбука
 - Във всеки ред се съдържа по една буква от дешифрираното съобщение, като тази буква е най-малката (според ASCII кода), която не се среща в реда и е по-голяма от най-малката буква в самия ред. Ако в реда се съдържат всички големи и малки латински букви то дешифрираният символ е ' '.

Програмата да изведе в конзолата дешифрираното съобщение.

Пример:

Вход	Исход

Пояснения:

1. Счита се, че при въвеждането на данните, потребителят ще спазва типа им. Ако не е упоменато друго, всички програми да извеждат в конзолата -1 ако входните данни, подадени от потребителя са извън дефиниционното множество на задачата и да прекратяват изпълнението си.
2. Всички задачи носят по 2 точки.
3. **За решаване на задачи 1, 2, 4 и 5 не се допуска използването на *string*, *cstring* и/или *vector*. Всички задачи трябва да бъдат решени чрез стандартни масиви от символи (*char[]*).**
4. 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.
5. 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.
6. Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC
7. Всяка задача от домашното трябва да бъде решена в точно един, отделен файл. Името на файла трябва да бъде в следния формат:

fnXXXXX_d3_N_CC.cpp, където:

- XXXXX 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++.
8. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:

UP_20-21_fnXXXXX_d3.zip, където XXXXX е вашият факултетен номер

9. 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.
10. Всички предадени програми трябва следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата.
11. 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!).
12. 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.

/ **

*

*