

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2021-2022 г.

Задачи за домашно № 3

1. Дадена последователност от думи ще наричаме правилна, ако всяка следваща дума се получава от предходната чрез точно една от следните трансформации:
 - Промяна на точно една буква;
 - Добавяне или премахване на точно една буква.

Да се напише функция *bool isCorrect(const char sequence[][20], int sequenceSize)*, която проверява дали определена последователност от думи е правилна. Да се напише *main()* функция, в която от клавиатурата се въвежда число $N \in [2, 50]$ - брой на думите, и самата последователност от думи и се извиква написаната функция върху така въведените данни. За целта на задачата дума дефинираме като последователност от малки латински букви с дължина между 1 и 19 символа. В зависимост от резултата от изпълнението на функцията, програмата да извежда на екрана *true* или *false*.

Пример:

Вход	Изход
4 bear beard heard hard	<i>true</i>
3 care tear wear	<i>false</i>

2. Да се напише функция, *int longestDist(const int [] arr, const int arrSize)*, която връща като резултат от изпълнението си индекса **на първото срещане** на елемента на масива, който има двете най-отдалечени срещания в масива.
 - При невалиден размер на масива ($arrSize < 2$), функцията да връща резултат -2
 - Ако всички елементи в масива имат различни стойности, функцията да връща резултат -1

- При повече от един елемент, отговарящ на условието, функцията да върне индекса на елемента, който се намира на *по-малък индекс в масива*.

Да се напише функция *main*, в която програмата получава от стандартния вход размер на масива $1 \leq N \leq 100$ и *N* на брой цели числа - елементи на масива, извиква функцията *longestDist* върху масива и извежда резултата на стандартния изход.

Пояснение: *arrSize* е с максимален размер 100.

Пример:

Вход	Изход	Обяснение
1 1	-2	Невалиден размер на масива
2 1 2	-1	Няма еднакви елементи в масива
2 3 3	0	3 е елементът с най-голямо разстояние между срещанията. Първото срещане на 3 в масива е на индекс 0.
3 5 2 5	0	5 е елементът с най-голямо разстояние между срещанията. Първото срещане на 5 в масива е на индекс 0.
5 33 11 22 11 22	1	11 е <u>най-левият</u> елемент с най-голямо разстояние между срещанията. Първото срещане на 11 в масива е на индекс 1.
4 6 7 7 6	0	6 е елементът с най-голямо разстояние между срещанията. Първото срещане на 6 в масива е на индекс 0.

- Ани се страхува от това, да не и бъде хакнат някой от профилите в социалните мрежи, затова решава да направи генератор за пароли, които да бъдат достатъчно сигурни. Вашата задача е да и помогнете да напише програма, която ще генерира тези пароли, разделени една от друга от знака "|".

Да се напише програма, която генерира серия от символи като в **шаблона**:

ABxyBA

като при всяко генериране на нов код, стойностите на символите се увеличават с

1. Ако **A** **надхвърли** 55, се връща на 35. Ако **B** **надхвърли** 96, се връща на 64.

От конзолата се чете 1 ред:

- На първия ред **a** – **цяло число** в интервала [1 ... 1000]
- На втория ред **b** – **цяло число** в интервала [1 ... 1000]
- На третия ред **максимален брой генерирани пароли** – **цяло число** в интервала [1 ... 1000000]

Ограничения:

- **A** е символ с ASCII стойност в диапазона [35... 55]
- **B** е символ с ASCII стойност в диапазона [64 ... 96]
- **x** е цяло число в диапазона [1... a]
- **y** е цяло число в диапазона [1... b]

Да се отпечата на конзолата генерираният код. Ако броят на комбинациите е по-голям от максималния на кода, да се отпечата до подадената стойност, в противен случай да се отпечата до текущия брой на комбинациите.

Пример:

Вход	Изход
2 3 10	#@11@# \$A12A\$ %B13B% &C21C& 'D22D' (E23E(
20 50 10	#@11@# \$A12A\$ %B13B% &C14C& 'D15D' (E16E()F17F) *G18G* +H19H+ ,I110I,

4. Да се напише функция `int longestSubstrWithUniqueSymbolsLength(const char str[])`, която по подаден като параметър символен низ връща дължината на най-дългия му подниз, който не съдържа повтарящи се символи. Да се напише `main()` функция, в която от клавиатурата се въвежда символен низ с максимална дължина 100 символа, състоящ се от малки латински букви, и се извежда на екрана резултатът от изпълнението на написаната функция върху него.

Пример:

Вход	Изход	Обяснение
abcccd	3	<i>abc</i>
aabbbbc	2	<i>ab</i> или <i>bc</i>
aaa	1	<i>a</i>

5. Да се напише програма, която въвежда от клавиатурата символен низ **X** с най-много 255 символа. Нека низът *s* е съставен само от малките латински букви на **X** в реда на срещането им в **X**, а низът *S* - само от главните латински букви на низа **X** в реда на срещането им в **X**. Програмата да изписва на стандартния изход "Yes", ако низовете *s* и *S* се състоят от съответни една на друга букви. В противен случай, програмата да изписва "No".

Пример:

Вход	Изход
------	-------

Последна промяна: 14.12.2021 г.

abABcC	Yes
aBACbc	No

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 и cstring. Всички задачи трябва да бъдат решени чрез стандартни масиви от символи (char[]).
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_d3_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_d3.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 3
* 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 3, компилирана с GCC, трябва да изглежда така:

```
/**
 *
 * Solution to homework assignment 3
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia
 University
 * Winter semester 2021/2022
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 3
 * @compiler GCC
 *
 */
```

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