

# СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

## Увод в програмирането, 2020-2021 г.

### Задачи за домашно № 2

1. Дадена редица от  $N$  на брой естествени числа, където  $N \in [3, 100]$ , се нарича *триъгълна*, ако абсолютните стойности на разликите между съседните ѝ елементи са огледално равни. Т.е. разликата между първия и втория елемент е равна на разликата между последния и предпоследния елемент; разликата между 2 и 3 елемент е равна на разликата между предпоследния и пред-предпоследния елемент; и т.н (ако редицата има нечетен брой елементи, то средния елемент участва в две разлики). Да се напише програма, която приема като вход  $N$  и редица от  $N$  естествени числа и извежда на екрана дали тя е *триъгълна*.

Вход	Изход
8 9 8 3 4 2 1 6 5	1
5 5 5 0 5 5	1
5 1 2 3 6 8 7	0

2. Да допуснем, че цените на акциите на компанията "Smashing Pyramid" са известни предварително за предстоящ период от  $N$  дни. Да се напише програма, която приема естествено число  $N \in [3, 365]$  и  $N$  на брой дробни числа, всяко от които представя цената на акцията за деня  $i$ , която принадлежи на интервала  $[0.1, 100]$ . Програмата да намери и изведе максималната достижима печалба от акциите на компанията "Smashing Pyramid", постигната чрез неограничен брой купувания и продавания в рамките на дадения период от  $N$  дни.

**Ограничение:** Потребителят има право да притежава най-много една акция в даден момент.

Вход	Изход	Пояснение
7 1 2 3 4 3 2 1 5	7	Купува се за 1 през 1-вия ден, продава се за 4 на 4-тия ден (печалба 3). Купува се пак за 1 на 7-мия ден и се продава за 5 на 8-мия (печалба 4). Общо печалба 7
5 15 14.5 10 7.45 5	0	Няма как да се реализира печалба.

3. Да се напише програма, която приема естествено число  $N \in [3, 1000]$  и  $N$  на брой арабски цифри. Програмата да извежда броя на всички подредици образувани от въведените цифри, които имат дължина минимум 3, и образуват палиндром. Ако няма подредици отговарящи на условието, да се изведе съобщение за грешка -1.

**Пояснение:** Подредица е всяка последователност с дължина  $k \leq N$ , която запазва реда на въведените числа (без да се пропуска някои от тях).

Вход	Изход	Пояснение
4 1 2 2 1	1	1221
5 5 0 6 0 5	2	060 50605
8 1 3 8 3 1 4 5 4 1 4	5	383 13831 454 14541 414

4. Да се напише програма, която приема цяло число  $N \in [3, 100]$  и редица от  $N$  на брой цели числа в интервала  $[-2147483648, 2147483647]$ . Програма да игнорира вече въведени числа, които се съдържат в редицата и да извежда като резултат броя на всички нулеви суми, които са съставени от произволни комбинации от 3 от въведените числа.

**Пример:**

Вход	Изход	Пояснение
3 1 2 3	0	Няма числа, такива, че сумата им да е равна на нула
3 1 1 1 1 1 1 1 2 1 1 1 -3	1	$1+2+(-3) = 0$
8 -1 0 1 1 2 3 4 3 -1 -2	2	$-1 + 0 + 1 = 0$ $-1 + 0 + 2 = 0$ $-2 + (-1) + 3 = 0$
7 -3 -2 -1 0 1 2 3	4	$-3 + 1 + 2 = 0$ $-3 + 0 + 3 = 0$ $-2 + 0 + 2 = 0$ $-1 + 0 + 1 = 0$

### Пояснения:

1. Всички задачи носят по 2,5 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 символни низове.
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:

fnXXXXX\_d2\_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++.
7. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:

UP\_20-21\_fnXXXXX\_d2.zip, където XXXXX 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 следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата.
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 2
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2020/2021
 *
 * @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 2, компилирана с GCC, трябва да изглежда така:

```

/**
 *
 * Solution to homework assignment 2
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2020/2021
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 2
 * @compiler GCC
 *
 */

```

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