

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2021-2022 г.

Задачи за домашно № 1

1. Да се въведе цяло естествено число n в конзолата, принадлежащо на интервала $[1...9]$ и да се отпечата на конзолата следната поредица от числа (празните водещи символи са интервали):

Пример:

Вход	Изход
5	1 2 3 4 5 2 3 4 5 3 4 5 4 5 5 5 4 5 4 3 5 4 3 2 5 4 3 2 1

2. Дадено е естествено число N $[1...10^9]$. Последователно се сравняват симетрично разположените спрямо средата му цифри, т.е. първата с последната, втората с предпоследната и т.н. При всяко такова сравнение по-голямата цифра “печели” - ако двете сравнявани цифри са равни, то и двете “губят”, а цифра, която няма с какво да бъде сравнена по подразбиране “печели”. Да се напише програма, която по въведено от клавиатурата число N извежда на конзолата печелившите цифри (в реда на извършените сравнения).

Пример:

Вход	Изход
583241	5 8 3
13257	7 5 2

3. Шеф на компания забелязва че все повече служители прекарват време в сайтове, които ги разсейват. За да предотврати това, той въвежда изненадващи проверки на отворените табове на браузъра на служителите си. Според сайта се налагат различни глоби:

- 'F' → 150 лв. (facebook.com)
- 'I' → 100 лв. (instagram.com)
- 'R' → 50 лв. (reddit.com)

От конзолата се четат два реда:

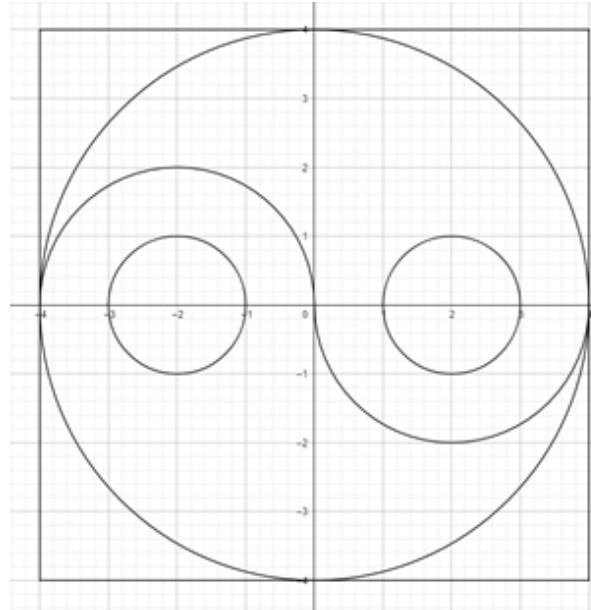
- Брой отворени табове в браузъра n - цяло число в интервала [1...10]
- Заплата - число в интервала [700...1500]
- След това n – на брой пъти се чете име на уебсайт – буква (d - уебсайт за програмиране без глоба)

Ако по време на проверката заплатата стане по-малка или равна на 0 лева, на конзолата се изписва "*You have lost your salary.*" и програмата приключва. В противен случай след проверката на конзолата се изписва остатъкът от заплатата (да се изпише като цяло число).

Пример:

Вход	Изход
10 750 <i>F</i> <i>I</i> <i>d</i> <i>F</i> <i>R</i> <i>F</i> <i>F</i>	You have lost your salary.
3 700 <i>d</i> <i>d</i> <i>d</i>	700
3 700 <i>F</i> <i>d</i> <i>d</i>	550

4. Да се напише програма, която прочита от клавиатурата координатите на точка в декартовата координатна система (две дробни числа, отделени с интервал) и определя дали точката се намира в черната, в бялата област на фигурата, дали е на границата между двете области или се намира извън фигурата. Програмата да извежда в конзолата съответно: „White”, “Black”, “Undefined” или „Outside”.



Пример:

Вход	Изход
-4.1 0.0	Outside
-4.0 -3.9	Undefined
-3.9 -3.9	White
3.9 3.9	White

5. Въвежда се естествено число n . Да се отпечата това естествено число без четните цифри. Ако в числото всички цифри са четни, да се отпечата 0.

Пример:

Вход	Изход
12345	135
223441	31
248	0
0	0

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 използване на масиви и STL функции.
5. Предадените от вас решения трябва да са написани на езика C/C++ и 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_d1_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_21-22_fnXXXXX_d1.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. При некоректен потребителски вход, програмата трябва да извежда съобщение "-1".
 10. Решенията трябва да се придържат към формата на потребителски вход/изход, както е указано в примерите, тъй като решенията ще бъдат проверени с автоматични тестове.
 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.

/* *

*

```

* Solution to homework assignment 1
* 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 2, компилирана с GCC, трябва да изглежда така:

```

/**
*
* Solution to homework assignment 1
* Introduction to programming course
* Faculty of Mathematics and Informatics of Sofia
University
* Winter semester 2021/2022
*
* @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.