

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2019-2020 г.

Задачи за домашно № 1

1. Да се напише програма на C++, която въвежда от клавиатурата цяло число в интервала $[-2^{31}, 2^{31}-1]$ ¹ и извежда на екрана отговор на въпроса дали в числото има **поне** две еднакви цифри.
2. Да се напише програма на C++, която въвежда от клавиатурата цяло число в интервала $[0, 1\,000\,000]$, което да се интерпретира като брой дни. Програмата да извежда на екрана на конзолата годината от Григорианския календар, която съответства на този брой дни. (забележка: при решаване на задачата да не се използва библиотеката *chrono*, както и да се вземат предвид високосните години²)

Пример:

Вход (брой дни)	Изход (година)
450527	1234

3. Да се напише програма на C++, която пресмята въведена от потребителя двуместна аритметична операция върху **положителни** едноцифрени шестнадесетични числа, като първо се въвежда операцията (+, -, *, /, %), след което се въвеждат последователно двата операнда. Резултатът да се изведе на конзолата. (За решаване на задачата не е позволено да се използва `std::hex`).

Пример:

Вход	Изход
* A 3 (aka 10*3)	1E
- B A	1
% C 8	4

¹ Помислете сами с кой от познатите ви прости типове данни, може да представите числата в посочения интервал.

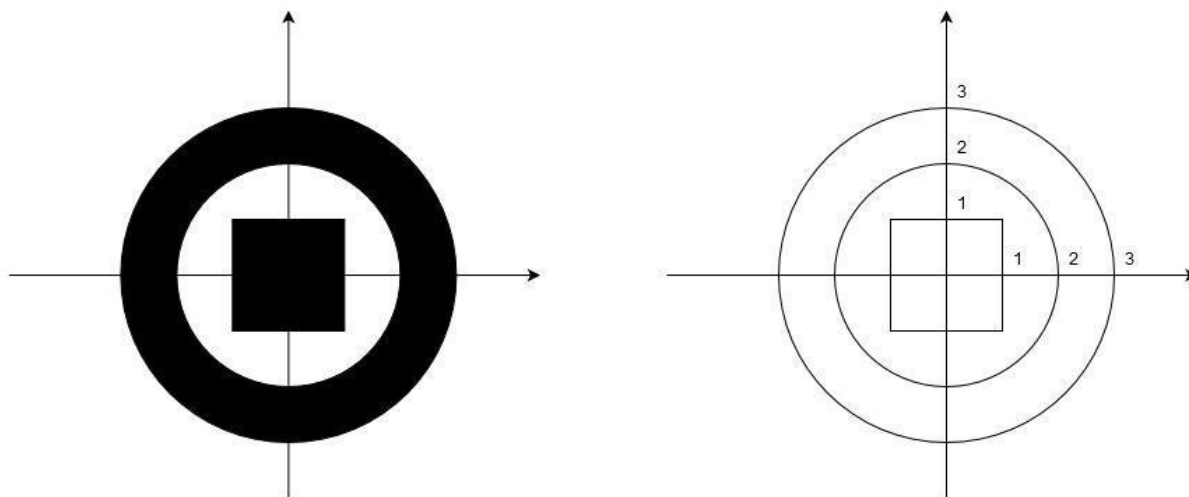
² Година, делища се на числото 4 без остатък, е високосна; ако годината обаче също се дели без остатък на 100, то тя не е високосна; но ако годината също се дели без остатък и на 400, то тя пак ще е високосна.

4. Да се напише програма, която реализира таймер. Потребителят въвежда число в интервала $[1, 2^{16}-1]$, след което програмата започва обратно броене от стойността на въведеното число до 0, като извежда на екрана оставащото време. **За решаването на задачата не се допуска използването на `sleep()`.**

Забележка: Опитайте се да реализирате времевия интервал чрез цикъл, който извършва някаква по-сложна операция. **При решението на задачата не се очаква да постигнете точност от гледна точка на реални единици за измерване на времето (секунди, минути и т.н.). Достатъчно е да се вижда ясно работата на брояча.** Препоръчително е да извеждате новите стойности на брояча, на едно и също място на екрана. За целта може да използвате `system("cls")`.

5. Да се напише програма на C++, която решава биквадратно уравнение от вида $ax^4+bx^2+c = 0$, където потребителя въвежда параметрите a , b и c . Ако уравнението има и корени, които не са реални числа, да се извежда подходящо съобщение за това.

6. 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 (x y)	Изход
-2 1	Black
0.5 1.3	White
0 2	Undefined

6 6	Outside
-----	---------

Пояснения:

1. Задачи 1-2 носят по 1 точка, а задачи 3-6 – по 2 точки
2. В решенията на дадените задачи **не се допуска** използването на оператора `goto`, `string` и/или масиви, както и на STL.
3. 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.
4. Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC
5. 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++.
6. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:

UP_19-20_fnXXXXX_d1.zip, където XXXXX е вашият факултетен номер

7. 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.
8. Всички предадени програми трябва да се държат адекватно при некоректни входни данни от потребителя. (например въвеждане на поредица от символи, когато програмата очаква число).
9. 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!).

10. 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 своите лични данни. За улеснение, просто копирайте дадения по-долу блок и попълнете в него нужната информация. 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 2019/2020
 *
 * @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 2019/2020
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 2
 * @compiler GCC
 *
 */
```

11. Предадени домашни, които не отговарят на условията от точки 2-10 ще бъдат оценени с 0 точки.