



Ученическо състезание „ЗАБАВНИТЕ ТЕХНОЛОГИИ” 2017

ПРИМЕРНИ ЗАДАЧИ ПО ПРОГРАМИРАНЕ

Задача 1. Голяма търговска верига решава да повиши оборота си и да привлече нови клиенти. Започва широка медийна кампания с обещание за големи намаления и резултатът не закъснява. Пред магазините се извиват дълги опашки - мечтата на търговеца. Охраната пуска по не повече от 20 клиенти да влизат или да излизат на всеки 5 минути. Да се състави програма (C++), чрез която се въвеждат цели числа (брой влезли или излезли клиенти) в интервала $[-20..20]$, като за брой едновременно влезли се въвежда положителна стойност, а за излезли клиенти - отрицателна стойност. Ако при пускане никой не е влязъл, или никой не е излязъл, не се въвежда нула.

Например портиерът отваря вратата и отвън има 25 човека, а вътре никой не чака да излезе. Тогава той пуска 20 души, 5 остават да чакат и в интерфейса на програмата се въвежда числото 20, а нулата за излезли не се въвежда. Ако при следващото пускане чакат за влизане 5 души и за излизане един, то в програмата ще се въведат числата 5 и -1.

Програмата да изчислява поотделно средния брой на влезлите и средния брой на излезлите за определен интервал от време клиенти (за край на въвеждането се въвежда 0).

Задача 2. Приемаме, че дадено 4-цифрено число е щастливо, ако левите му 2 цифри съвпадат с десните му 2 цифри. Например, 1414 е щастливо число. Да се състави програма, която въвежда чрез клавиатурата 50 естествени числа от интервала $[1000 .. 9999]$ и преброява колко от тези числа са щастливи. При опит за въвеждане на числа извън този интервал те да се игнорират, потребителят да бъде уведомен, че трябва да въвежда други подходящи числа на тяхно място.

Критерии за оценяване

За частична реализация на задачата могат да се дадат до 30 т. по преценка на журито и в зависимост от това каква част е реализирана.

За реализация и тестване на основния алгоритъм на задачата се дават 60 точки.

За изследване и предотвратяване на всички възможни проблемни ситуации (например деление на нула, въвеждане на некоректни данни, гранични случаи) се дават 30 точки.

За добър стил на програмиране (използване на масиви, функции, добра програмна структура) се дават 10 точки.