АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
КРАТНИ ПАЛИНДРОМИ

Ограниченията в задачата изискват прецизиране на решението. Очевидно, има два конкурентни подхода за решаване: в интервала да проверяваме кратните за „палиндромичност“ или палиндромите за кратност. Естествено е да реализираме и двата алгоритъма и да изберем „по-евтиния“ за зададените входни данни. n-цифрените числа са $9\times10^{n-1}$. Кратните на p са в порядъка $[9\times10^{n-1}/p]$. Палиндромите са еднозначно определени от първата си „половина”, например, така че техният брой се оценява като $9\times10^{[n/2]-1}$ (или 10 пъти повече, ако има „средна цифра”, т.е. n е нечетно). Записано с обща формула: $10^{(n \bmod 2)}\times9\times10^{[n/2]-1}$.
Проверката за палиндром на кратните на p може да става по „стандартен” начин (обръщане на числото и проверка дали обратното съвпада с даденото), но този алгоритъм изисква на всяка проверка n деления. Те могат да се намалят наполовина, ако обръщаме само „втората половина” и проверяваме дали полученото е равно на „първата половина”. Още по-добре би било, ако проверяваме цифрите „от двата края” и спираме при първо несъвпадение, давайки отрицателен отговор. За да намалим деленията в този случай е добре да се ползва функцията lldiv от стандартната библиотека, която е оптимизирана за получаването на частното и остатъка с едно деление.
Аналогично можем да подходим при създаването на палиндром „от две половини” – вместо да обръщаме едно число на всяка стъпка, организираме „обратно увеличаване с 1” на втората част, което може да се постигне дори без деление и изисква малко действия. По такъв начин идеята за тестване на палиндромите за кратност печели още пространство (около 4 пъти по-бърза проверка).
В повечето случаи на работа с цифри е оправдано предварително да се създаде масив с необходимите степени на десетката.

Автор: Павлин Пеев