

- I. Utwórz statyczną funkcję `splitToDigits` przyjmującą jako argument wartość całkowitą dodatnią i zwracającą jednowymiarową tablicę wartości całkowitych dodatnich. Celem funkcji będzie utworzenie tablicy o rozmiarze tożsamym z ilością cyfr liczby dostarczonej jako argument, a następnie wypełnienie jej cyframi z jakich składa się dostarczona liczba.
- II. Utwórz statyczną funkcję `isArmstrongNumber` przyjmującą jako argument wartość typu `int` i dostarczającą jako rezultat wartość logiczną. Przyjmij że działanie tej funkcji będzie adekwatne do nazwy funkcji i będzie sprawdzało czy dostarczona jako argument wartość jest liczbą *Armstronga* (*narcystyczną*).
- III. Dany jest nagłówek funkcji:

```
1 public static int[][] calculateSquares(int screenWidth, int ↵
    screenHeight, int side)
```

przyjmujący jako argumenty odpowiednio: szerokość i wysokość ekranu, oraz długość boku kwadratu. Uzupełnij ciało tej metody algorytmem wyliczającym ile kwadratów o zadanej długości boku można umieścić na ekranie. Następnie wylicz współrzędne (górnego lewego rogu) x i y wszystkich kwadratów i zwróć je w postaci dwu wymiarowej tablicy w której pierwsza kolumna opisuje współrzędne x a druga y .
- IV. Utwórz i przetestuj metodę rekurencyjną, która stwierdzi czy dostarczone w tablicy zmiennych typu `char` słowo jest palindromem.
- V. Przekształć metodę sprawdzającą czy tablica `char[]` reprezentuje palindrom w ten sposób, aby na ekranie wypisany został tekst od środka, np dla słowa *Anna* oczekujemy *nAan*. Jeżeli palindromem ma nieparzystą ilość znaków, wówczas należy pominąć znak środkowy, zatem dla słowa *kajak* uzyskamy *akka*.
- VI. Utwórz rekurencyjną metodę, która wypełni dowolną kwadratową tablicę zmiennych typu `int` w taki sposób, aby kolejne wartości liczbowe na kolejnych wewnętrznych pierścieniach zwiększały się o jeden np:

```
1 11111111
2 12222221
3 12333321
4 12344321
5 12344321
6 12333321
7 12222221
8 11111111
```