

# Laboratorium AiSD

## Lista 1

Proszę pamiętać, że **część rozwiązania** zadania stanowi również **zestaw testów** zaimplementowanych algorytmów i/lub struktur danych. Dodatkowo, proszę zwracać uwagę na **powtarzające się fragmenty** kodu i wydzielać je do osobnych funkcji/klas.

1. Zaimplementuj i przetestuj poniższe procedury uwzględniając kryteria:

- Każda figura ma być rysowana **bezpośrednio w konsoli** – nie dopuszcza się korzystania z tablic/łańcuchów znaków wielokrotnie modyfikowanych, a następnie wyświetlanych na ekranie,
- Powtarzające się lub logicznie osobne funkcjonalności/fragmenty programu należy wydzielić do osobnych funkcji pomocniczych.

a. `void drawZigZag(int n, int l)` wyświetlającą na ekranie zygzak o wysokości  $n$  linii i długości linii  $l$ , na przykład:

1	1 = 5	18	n = 6
2	n = 0	19	XXXXX
3		20	XXXXX
4	n = 1	21	XXXXX
5	XXXXX	22	XXXXX
6		23	XXXXX
7	n = 2	24	XXXXX
8	XXXXX	25	...
9	XXXXX	26	
10		27	
11	n = 3	28	n = 11
12	XXXXX	29	XXXXX
13	XXXXX	30	XXXXX
14	XXXXX	31	XXXXX
15		32	XXXXX
16	...	33	XXXXX
		34	XXXXX
		35	XXXXX
		36	XXXXX
		37	XXXXX
		38	XXXXX
		39	XXXXX

b. `void drawScarf(int n, int k)` wyświetlającą na ekranie „szalik” będący zygzakiem o wysokości  $n$  i szerokości linii  $2k + 1$  wzbogaconym wzorkiem:

1	k = 3	18	n = 11
2	n = 0	19	XXXOXXX
3		20	XXOOOXX
4	n = 1	21	XOOOOOX
5	XXXOXXX	22	XXOOOXX
6		23	XXXOXXX
7	n = 2	24	XXOOOXX
8	XXXOXXX	25	XOOOOOX
9	XXOOOXX	26	XXOOOXX
10		27	XXXOXXX
11	n = 3	28	XXOOOXX
12	XXXOXXX	29	XOOOOOX
13	XXOOOXX		
14	XOOOOOX		
15			
16	...		

2. Napisz funkcję `int[][] longestNondecreasingSubstrings(int[] numbers)`, która przechodzi po tablicy `numbers` i zwraca tablicę kolejnych, najdłuższych niemalejących podciągów liczb.

Pisząc kod zastanów się jak sprawić, by po tablicy `numbers` przejść dokładnie raz.

W zadaniu można korzystać **jedynie** z podstawowych tablic dostępnych w języku JAVA. W szczególności, **zabronione** jest korzystanie z kolekcji takich jak `ArrayList`.

Przykład:

Wejście:

{ 1, 2, 8, -3, -3, -4, 7, 10, 0 }

Wyjście:

```
{
  { 1, 2, 8 },
  { -3, -3 },
  { -4, 7, 10 },
  { 0 }
}
```