

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 drawBridge(int n)` wyświetlającą na ekranie most następującej postaci:

1	n = 0	20	
2		21	n = 5
3	n = 1	22	I I
4	X	23	I \ / I
5		24	I \ / I
6	n = 2	25	XXXXXXXXXX
7	XXX	26	I I
8	I I	27	
9		28	n = 6
10	n = 3	29	I I
11	I I	30	I \ / I
12	XXXXX	31	I \ / I
13	I I	32	I \ / I
14		33	XXXXXXXXXXXX
15	n = 4	34	I I
16	I I		
17	I \ / I		
18	XXXXXXXX		
19	I I		

2. Napisz funkcję `int[][] longestGrowingSubstrings(int[] numbers)`, która przechodzi po tablicy `numbers` i zwraca tablicę kolejnych, najdłuższych rosnących 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 }
}
```