Programowanie funkcyjne i współbieżne
Lista 1

Wszystkie programy mają być napisane w stylu funkcyjnym z użyciem rekursji.
W poniższych zadaniach dla list wolno używać tylko funkcji head i tail!
Po definicji każdej funkcji należy umieścić wyczerpujący zbiór testów (patrz suma).

1. Napisz funkcję suma: List[Double] => Double, zwracającą sumę liczb z podanej listy,
   np. suma(Nil) == 0.0
   suma(List(-1, 2, 3)) == 4.0
   suma(List(5.6)) == 5.6

2. Napisz funkcję ends[A](xs: List[A]): (A, A), zwracającą parę, zawierającą pierwszy i ostatni element zadanej listy,
   np. ends(List(1, 3, 5, 6, 9)) == (1,9)
   ends(List("Ala", "ma", "kota")) == ("Ala", "kota")
   ends(List(1)) == (1,1)
   ends(Nil) =>> wyjątek NoSuchElementException: empty list

3. Napisz funkcję posortowana: List[Int] => Boolean sprawdzającą, czy dana lista jest posortowana niemalejąco,
   np. posortowana(List(1,3,3,5,6,7)) == true

4. Napisz funkcję glue: (List[String], String) => String, która na wejściu przyjmuje listę napisów oraz napis będący separatorem i zwraca napis będący połączeniem napisów wejściowych oddzielonych od siebie zadanym separatorem,
   np. glue(List("To", "jest", "napis"), "-") == "To-jest-napis"
   glue(Nil, "-") == ""