

01. Podstawy

Zad 1.1

Napisz kod, który dla dowolnej tablicy liczb wyznaczy:

- A. liczbę elementów dodatnich
- B. sumę elementów, które po zaokrągleniu są podzielne przez 3 - policz zarówno sumę elementów przed jak i po zaokrągleniu
- C. odległość każdego z elementów od liczby 8
- D. jego postać znormalizowaną - minimum przechodzi na -1, maks na 1, a pozostałe elementy mają zostać liniowo przeskalowane według poniższego wzoru, gdzie min, max to ekstrema wektora, a new max/min to docelowy zakres - w naszym przypadku -1 oraz 1
$$x' = \frac{x - \min}{\max - \min} \cdot (\text{new}_{\max} - \text{new}_{\min}) + \text{new}_{\min}$$
- E. średnią wartość kwadratów liczb większych od 5 lub mniejszych od 2
- F. jej średnią
- G. jej wariancję
- H. jej minimum i maksimum
- I. tablicę o połowę krótszą, która na pierwszym miejscu będzie mieć sumę elementów pierwszego i ostatniego, na drugim drugiego i przedostatniego... na ostatni sumę dwóch środkowych (jeśli długość parzysta) lub podwojony środkowy element (jeśli długość nieparzysta)

Zad 1.2

Dla dwóch tablic równej długości x i y oblicz ich korelację ze wzoru:

A.
$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

B.
$$r = r_{xy} = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sqrt{(\sum x_i^2 - n \bar{x}^2)} \sqrt{(\sum y_i^2 - n \bar{y}^2)}}.$$

C.
$$r = r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}.$$

Zad 1.3

Dla danej listy, zwróć listę zawierającą tylko unikatowe wartości (bez użycia zbiorów - Set).

Zad 1.4

Korzystając ze wzoru Leibniza $\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \pi/4$ oblicz przybliżoną wartość liczby pi dla

1 000, 10 000 i 100 000 początkowych wyrazów i porównaj uzyskane liczby z wartością dostępną w stałej `Math.pi`.

Zad 1.5

Podana wyżej metoda nie jest jedyną na przybliżanie liczby pi. Skorzystamy teraz z metody Monte Carlo, której algorytm wygląda następująco:

1. Wylosuj n punktów z dwuwymiarowej przestrzeni $[-1,1] \times [-1,1]$
2. Sprawdź ile punktów jest oddalonych od punkt (0,0) o mniej niż 1
3. Podziel tę liczbę przez n i przemnoż przez 4

Do losowania punktów użyj metody `nextDouble()` z klasy `Random`

Zad 1.?

<https://projecteuler.net/problem=1>

<https://projecteuler.net/problem=2>

itd...

Zachęcam do założenia konta i wysyłania odpowiedzi.

codewars.com

Multiply

<https://www.codewars.com/kata/50654ddff44f800200000004>

Return negative

<https://www.codewars.com/kata/return-negative/java>

Opposite numbers

<https://www.codewars.com/kata/56dec885c54a926dcd001095>

Even or odd

<https://www.codewars.com/kata/53da3dbb4a5168369a0000fe>

Find odd cubes

<https://www.codewars.com/kata/580dda86c40fa6c45f00028a>

Ones and zeroes

<https://www.codewars.com/kata/578553c3a1b8d5c40300037c>

Counting sheep

<https://www.codewars.com/kata/counting-sheep-dot-dot-dot>

MakeUpperCase

<https://www.codewars.com/kata/57a0556c7cb1f31ab3000ad7>

Perfect Square

<https://www.codewars.com/kata/56269eb78ad2e4ced1000013>

Count of positives / sum of negatives

<https://www.codewars.com/kata/count-of-positives-slash-sum-of-negatives>

Jenny's secret message

<https://www.codewars.com/kata/jennys-secret-message/java>

Sum without highest and lowest number

<https://www.codewars.com/kata/sum-without-highest-and-lowest-number>

Find the first non-consecutive number

<https://www.codewars.com/kata/58f8a3a27a5c28d92e000144>

Multiples of 3 and 5

<https://www.codewars.com/kata/514b92a657cdc65150000006>

Rock paper scissors

<https://www.codewars.com/kata/5672a98bdbdd995fad00000f>

Closest to zero

<https://www.codewars.com/kata/closest-to-zero/train/java>

02. Instrukcje sterujące

Zad 2.1

Napisz metodę statyczną w klasie MathUtils która oblicza wartość bezwzględną dla:

- int
- double

Zad 2.2

Napisz metodę cumSum(int[] tab) liczącą skumulowaną sumę dla tablicy wartości liczbowych. Skumulowana suma dla tablicy to również tablica o takiej samej długości, gdzie i-ty element tablicy wynikowej to suma elementów od 1 do i-tego z tablicy wejściowej.

Dla tablicy:

[1, 3, 12, -3]

Skumulowana suma to:

[1, 4, 16, 13]

Zad 2.3

Napisz metodę dziesiętnaNaBinarna(int x), która powinna zwracać napis z binarnym zapisem przekazanej liczby.

Algorytm na zamianę liczby z systemu dziesiętnego (dowolnego) na binarny:

Wprowadźmy zmienną `liczba`, równą liczbie, którą chcemy zamienić na system binarny. Dopóki `liczba` nie jest równa zero, to wykonuj dwa polecenia: zapisz resztę z dzielenia, a następnie podziel `liczbę` przez 2 i zapisz wynik w zmiennej `liczba`.

Liczbę cyfr w systemie dwójkowym możemy z góry oszacować poprzez $\log_2(\text{liczba})$ (logarytm o podstawie 2). Dla jakiej liczby algorytm nie zadziała? (przypadek ten należy obsłużyć "ręcznie")

Zad 2.4

Napisz funkcję `zlicz`, która dla danej wartości całkowitej $k > 0$ i tablicy liczb całkowitych x o n elementach ze zbioru $\{1, 2, \dots, k\}$, zwróci tablicę o długości k , w której i -ty element jest równy liczbie wystąpień wartości i w x , dla $i = 1, \dots, k$.

Przykład:

```
zlicz([1,4,1,2,1], 5) == [3, 1, 0, 1, 0]
```

Bo były trzy jedynki, jedna dwójka, zero trójek, jedna czwórka i zero piątek.

Zad 2.5

Za Wikipedią: Sortowanie przez zliczanie (ang. counting sort) – metoda sortowania danych, która polega na sprawdzeniu ile wystąpień kluczy mniejszych od danego występuje w sortowanej tablicy. Algorytm zakłada, że klucze elementów należą do skończonego zbioru (np. są to liczby całkowite z przedziału $0..100$), co ogranicza możliwości jego zastosowania.

Korzystając z funkcji napisanej w poprzednim zadaniu (`zlicz`) użyj jej aby wypisać posortowaną tablicę powyższą metodą. Skoro wiadomo, że były trzy 1, jedna 2, zero 3 itd to oznacza, że wystarczy wypisać wektor postaci: $[1, 1, 1, 2, 4]$.

Utrudnienie - funkcja nie musi przyjmować wartości k , a powinna ją sama wyliczać na podstawie wektora wejściowego.

Zad 2.6

Zaimplementuj funkcję, `is.prime(int x)`, która dla danego liczby naturalnej x zwraca wartość logiczną, która oznacza czy x jest liczbą pierwszą czy nie.

Tzw. formuła Eulera $n^2 + n + 41$ generuje 40 różnych liczb pierwszych dla n całkowitych od 0 do 39. Możesz użyć tych liczb do testów.

Zad 2.7

Napisz funkcję `join(int[] values, String separator)` która dla danej tablicy wartości oraz separatora, zwróci jeden napis ze wszystkimi wartościami złączonymi z separatorem pomiędzy każdą wartością. Dla tablicy o elementach 1, 3 oraz 6 i separatorze "@" należy zwrócić napis: "1@3@6".

Uwaga: nie należy doklejać separatora na początku ani na końcu napisu!