

## Zadanie 3.1

---

Zdefiniuj moduł **Math** i umieść w nim definicję funkcji **f** określonej następująco:

- dla  $x < -10$  wartość funkcji jest równa kwadratowi liczby  $x$ ,
- dla  $-10 \leq x < 0$  wartość funkcji jest równa sinusowi z liczby  $x$ ,
- dla  $0 \leq x \leq 2$  wartość funkcji jest równa pierwiastkowi z liczby  $x$ ,
- dla wartości większych niż 2 wartość funkcji nie jest określona (**Nie definiujemy jej dla  $x > 2$ .**).

załaduj moduł **Math** i przetestuj działanie funkcji **f**. Co się dzieje gdy zostanie ona wywołana z argumentem większym niż 2?

## Zadanie 3.2

---

Zdefiniuj funkcję **factorial** wyznaczającą silnię z liczby całkowitej nieujemnej  $n$ . Korzystając z funkcji **factorial** zdefiniuj funkcję **binomialCoefficient** wyznaczającą wartość symbolu Newtona dla dwóch nieujemnych liczb całkowitych  $k, n$ . Funkcje umieść w module **Math**.

## Zadanie 3.3

---

(\*) definiuj rekurencyjną funkcję **factorial2** wyznaczającą silnię podwójną z liczby całkowitej nieujemnej  $n$ :

- $n!! = 1$  dla  $n = 0$  i  $n = 1$
- $n!! = n(n - 2)!!$  dla  $n > 1$

Funkcję umieść w module **Math**.

## Zadanie 3.4

---

Zdefiniuj funkcję **divides**, która jako swoje argumenty przyjmuje dwie liczby całkowite  $k$  oraz  $n$  (typu `Int`) i zwraca wartość logiczną:

- **True** jeśli  $k$  dzieli  $n$ ,
- **False** w przeciwnym przypadku.

**Wskazówka:** W definicji funkcji **divides** można użyć funkcję **rem**.

Funkcję umieść w module **Math**.

## Zadanie 3.5

---

Zdefiniuj rekurencyjną funkcję **seq'**, wyznaczającą wartość  $n$ -tego wyrazu ciągu liczbowego zdefiniowanego następująco:

- $seq\_1 = 3$
- $seq\_2 = 4$
- $seq\_n = 0.5 * seq\_n - 1 + 2 * seq\_n - 2$ , dla  $n > 2$

Funkcję umieść w module **Math**.

## Zadanie 3.6

---

(\*) Zdefiniuj własną funkcję **gcd'** (największy wspólny dzielnik). Implementacja ma się opierać na algorytmie Euklidesa. Można stosować wyłącznie operatory relacyjne i operator odejmowania. Funkcję umieść w module **Math**.

## Zadanie 3.7

---

Zdefiniuj operator  $><$ , który przyjmuje jako argumenty dwie liczby całkowite  $a$  oraz  $b$  i zwraca wartość logiczną określającą czy liczby  $a$ ,  $b$  są względnie pierwsze. Operator umieść w module **Math**.

**Wskazówka:** Liczby  $a$  i  $b$  są względnie pierwsze jeżeli ich największym wspólnym dzielnikiem jest 1.

## Zadanie 3.8

---

(\*) Przyjmijmy, że liczba zespolona jest zapisywana jako para liczb typu **Double**. Zaimplementuj moduł **Complex** zawierający:

- operator dodawania liczb zespolonych (+.);
- operator odejmowania liczb zespolonych (-.);
- operator mnożenia liczb zespolonych (\*.);
- funkcje **re** i **im** zwracające część całkowitą i urojoną liczby zespolonej;
- funkcję **i** zwracającą jako wynik jednostkę urojoną.

## Zadanie 3.9

---

(\*) Przyjmijmy, że wektor w przestrzeni  $R^3$  jest zdefiniowany jako trójka liczb typu **Double**. Zaimplementuj moduł **Vectors** zawierający:

- operator dodawania wektorów (+.);
- operator odejmowania wektorów (-.);
- operator mnożenia skalarnego wektorów (\*.);
- operator mnożenia wektorowego wektorów (\*\*).