

Technologie Informacyjne i Komunikacyjne R Python

Zadania - poziom podstawowy

Instrukcje warunkowe, pętle, funkcje

Zad. 1

Poćwicz używanie Python'a jako kalkulatora (w sesji interakcyjnej) i policz 2^{23} , $\sin(\pi/8)$, resztę z dzielenia 17 przez 3, $\sqrt{3455}$. Do policzenia funkcji \sin , używania liczby π itp. niezbędne jest dodanie biblioteki matematycznej:

```
import math
```

Następnie możemy używać funkcji i stałych z tej biblioteki pisząc:

```
math.pi, math.sin(x) ...
```

Dokumentacja:

<https://docs.python.org/3/library/math.html>

Zad. 2

Napisz program wypisujący powitanie i informację co robi, a następnie proszący o wpisanie najpierw jednej, a potem drugiej liczby całkowitej i sprawdzający czy pierwsza dzieli się bez reszty przez drugą i wypisujący na ekran stosowny komunikat.

Zad. 3

Napisz program wczytujący trzy liczby (całkowite lub rzeczywiste) z klawiatury i wypisujący na ekran największą z nich. W programie mogą być wykorzystane tylko dwie instrukcje `if` bez `elif`, `else`, `or`, `and`.

Zad. 4

W pliku `funkcje.py` zdefiniuj trzy funkcje: liczące pole koła, trójkąta i kwadratu. Wewnątrz żadnej z funkcji nie powinno być instrukcji `print`. Stwórz drugi plik `poła.py`, który zaimportuje plik `funkcje.py`. Wczytaj w nim z klawiatury niezbędne dane. Wywołaj funkcje i ich wyniki wypisz na ekran.

Zad. 5

- Napisz program używający pętli `for` do liczenia sumy liczb całkowitych od 1 do 100
- W tym samym skrypcie dodaj drugą pętlę `for` liczącą sumę kwadratów liczb całkowitych od 5 do 10

Wyniki wypisz na ekran.

Zad. 6

Napisz program wyliczający pierwiastki równania kwadratowego. Program powinien różnie reagować (wypisywać różne komunikaty/wyniki) w zależności od obliczonej delty.

Ważne !! Sprawdź działanie programu dla $a=1$, $b=-0.2$ i $c=0.01$ (jedno miejsce zerowe $x=0.1$).

Zad. 7

Stwórz funkcję liczącą $n!$. Zadbaj, żeby prawidłowo było liczone $0!$. Korzystając z napisanej funkcji i pętli `for` wypisz $n!$ dla $n \in (0, 20)$. Używając przekierowania standardowego strumienia wyjściowego zapisz wynik do pliku `wyniki_silnia.txt`

Zad. 8

Liczba doskonała to liczba, która jest równa sumie swoich dzielników (razem z 1, bez samej siebie), np.

$$6 = 1 + 2 + 3$$

lub

$$28 = 1 + 2 + 4 + 7 + 14$$

Napisz funkcję, która sprawdzi czy podana liczba jest liczbą doskonałą i zwróci wartość logiczną `True` lub `False`. Użyj tej funkcji w programie, który:

- wersja 1: wypisuje na ekran prośbę o podanie liczby całkowitej, wczytuje ją, sprawdza czy jest doskonała i wypisuje odpowiedni komunikat na ekran

- wersja 2: znajduje i zapisuje do listy trzy pierwsze liczby doskonałe

Używanie bibliotek Numpy i Matplotlib

Może się przydać:

- `linspace(x_min, x_max, n)` - tworzy tablicę n równoodległych liczb z przedziału $[x_min, x_max]$
- `arange(x_min, x_max, dx)` - tworzy tablicę równoodległych (o dx) liczb; ostatnia liczba jest mniejsza od x_max

Zad. 9

Stwórz dwie macierze jednowymiarowe: macierz A, wypełnioną parzystymi liczbami całkowitymi z przedziału $[1,10]$ i macierz B, wypełnioną ujemnymi, parzystymi liczbami całkowitymi z przedziału $[-10,-1]$. Użyj funkcji `arange`!. Stwórz macierz C, której element o indeksie i jest iloczynem elementów o indeksie i w tablicach A i B. Wypisz na ekran liczby z macierzy A, B i C.

Zad. 10

Wczytaj do macierzy dane z pliku `~kaste/dane_python/funkcja1.dat`. Policz średnią liczb z pierwszej kolumny i średnią liczb z drugiej kolumny z pliku. Wypisz na ekran wynik w postaci:

Srednia liczb z pierwszej kolumny= xx

Srednia liczb z drugiej kolumny= yy

Zad. 11

W pliku `~kaste/dane_python/funkcja2.dat` zostały zapisane trzy kolumny liczb w formacie: x , y , błąd y . Wczytaj dane z pliku i narysuj najpierw same punkty, nie połączone linią, a potem punkty z błędami (również nie połączone linią). Opisz osie. Sprawdź jak zmienić rodzaj i kolor punktów i zmień je na czerwone trójkąty. Dodatkowo: na rysunku dodaj tekst napisany w LaTeX'u:

Punkty wygenerowane zgodnie ze wzorem $a * e^{-\frac{(x-m)^2}{2s^2}} + b * e^{-\frac{(x-n)^2}{2r^2}} + h$. Zapisz rysunek do pliku `dane.pdf`.

Zad. 12

- Wczytaj do macierzy dane z pliku `~kaste/dane_python/funkcja2.dat`.
- Dopasuj do punktów funkcję $f(x) = a * e^{-\frac{(x-m)^2}{2s^2}} + b * e^{-\frac{(x-n)^2}{2r^2}} + h$
- Przy dopasowaniu zwróć uwagę na to, żeby użyć błędów punktów σ_i (trzecia kolumna).
- Zadbaj o ustawienie początkowych wartości parametrów.
- Narysuj punkty z błędami (kolor zielony, duże kropki).
- Dodaj do rysunku dopasowaną krzywą (kolor czerwony).
- Wypisz na ekran wartości dopasowanych parametrów z opisem i błędami.
- Na rysunku dodaj opis osi
- Zapisz rysunek do pliku `fit.pdf`.

Zad. 13 Dla dopasowanej funkcji i danych z poprzedniego zadania policz wartość $\chi^2 / (\text{liczba_stopni_swobody})$ gdzie:

$$\chi^2 = \sum_{i=1}^n \frac{(y_i - f(x_i; a_1, \dots, a_l))^2}{\sigma_i^2}$$

a liczba stopni swobody to liczba punktów pomiarowych pomniejszona o liczbę dopasowywanych parametrów funkcji. Symbole a_i oznaczają parametry funkcji, a σ_i błędy punktów.

Zadania - poziom rozszerzony:

Zad. 1

Stwórz funkcję liczącą $n!$ metodą rekurencyjną (funkcja woła samą siebie). Przetestuj jej działanie.

$$\begin{cases} a_0 = 1 \\ a_n = a_{n-1} \cdot n \quad \text{dla } n > 0 \end{cases} \quad (1)$$

Zad. 2

Napisz funkcję liczącą metodą rekurencyjną n -ty element ciągu Fibonacciego zdefiniowanego w następujący sposób:

$$\begin{cases} a_1 = 1 \\ a_2 = 1 \\ a_n = a_{n-1} + a_{n-2} \quad \text{dla } n > 2 \end{cases} \quad (2)$$

Stwórz pustą listę, a następnie korzystając z napisanej funkcji i pętli `for` wypełnij listę pierwszymi dziesięcioma elementami ciągu Fibonacciego. Wypisz zawartość listy na ekran.

Zad. 3

Napisz program zapisujący do listy n pierwszych liczb pierwszych (n wczytywane z klawiatury), a następnie wypisujący zawartość listy na ekran. Postaraj się skorzystać z pełnej wersji instrukcji `for` (razem z `else`), tak żeby lista była rozbudowywana tylko wtedy gdy odpowiednia pętla nie zostanie przerwana instrukcją `break`.

Zad. 4

Napisz funkcję, która znajduje największy wspólny dzielnik dwóch liczb naturalnych a i b metodą Euklidesa i sprawdź jej działanie w programie. Skorzystaj z algorytmu:

- Jeżeli $a < b$, to zamień a i b
- Dopóki b jest większe od zera:
 - licz resztę z dzielenia (r) większej liczby (a) przez mniejszą (b)
 - zastąp a przez b
 - zastąp b przez r
- zwróć a

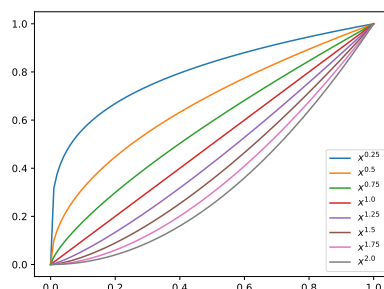
Zad. 5

Narysuj na jednym rysunku rodzinę funkcji Gauss'a

$f(x) = a * e^{-\frac{(x-m)^2}{2s^2}}$, które mają położenie maksimum (parametr m) w zerze, wysokości (parametr a) równe 10 i wartości parametru s (σ - miara szerokości rozkładu) z przedziału $[0.5, 5.5]$, zmieniające się co 0.5. Zakres osi x to: $[-20, 20]$. Sprawdź jak zmienić rodzaj linii i zmień je na przerywane.

Zad. 6

Narysuj na jednym rysunku rodzinę funkcji potęgowych $f_\alpha(x) = x^\alpha$ dla $x \in [0, 1]$, dla $\alpha \in \{0.25, 0.5, \dots, 2\}$ (użyj funkcji `linspace`). Sprawdź jak dodać legendę (sprawdź opcję `label` polecenia `plot`), tak żeby w legendzie (`plt.legend()`) przy każdej linii była odpowiednia postać funkcji: $x^{0.25}$ itd.



Zad. 7 Napisz program wypełniający podobnie jak w poprzednim zadaniu macierz n rzeczywistymi liczbami losowymi z przedziału od a do b , a następnie sortujący liczby w macierzy metodą bąbelkową.
Przykład działania algorytmu sortującego:

4352 → 3452 → 3452 → 3425

3425 → 3425 → 3245

3245 → 2345 → 2345

Algorytm:

- Zewnętrzna pętla powtarza się jeśli w tablicy któreś z elementów zostały zamienione (\Rightarrow może jeszcze coś trzeba będzie zamienić. Jeśli już nic nie było zamieniane, to znaczy że tablica jest uporządkowana.)
- Wewnętrzna pętla przechodzi przez wszystkie elementy tablicy i sprawdza czy i -ty element jest większy od elementu $i+1$. Jeśli tak, to zamienia elementy i ustawia zmienną logiczną na `True`.