

## Zestaw 2

1. Mamy zagnieżdżoną listę, która może zawierać różne heterogeniczne typy, na przykład inną listę, ale również krotkę, słownik. Dodaj element o kolejnej wartości w **najbardziej zagnieżdżonej** liście. Napisz program, który zrobi to uniwersalnie, dla dowolnego zagnieżdżenia, również jeśli pojawią się inne typy. Dla `[1 [2, 3] 4]` chodzi o `[1 [2, 3, 4] 4]`, dla `[3, 4, [2, [1, 2, [7, 8], 3, 4], 3, 4], 5, 6, 7]` powinno być `[3, 4, [2, [1, 2, [7, 8, 9], 3, 4], 3, 4], 5, 6, 7]`. Jeżeli największe zagnieżdżenie na danym poziomie się powtórzy, należy dodać w obu zagnieżdżeniach, czyli dla `[1, [3], [2]]` należy uzyskać `[1, [3, 4], [2, 3]]`. Przykład bardziej złożony: `list1 = [1, 2, [3, 4, [5, {'klucz': [5, 6], 'tekst': [1, 2]}], 5], 'hello', 3, [4, 5], (5, (6, (1, [7, 8])))]`. Tutaj na takim samym, największym poziomie zagnieżdżenia, są listy będące wartościami w słowniku (listy `[5, 6]`, `[1, 2]`) a także zagnieżdżona w krotkach (lista `[7, 8]`) i do to do nich powinien zostać dodany kolejny element. Zatem oczekiwane jest: `[1, 2, [3, 4, [5, {'klucz': [5, 6, 7], 'tekst': [1, 2, 3]}], 5], 'hello', 3, [4, 5, 6], (5, [6, [7, 8, 9]])]`.

2. Dla dowolnego podanego łańcucha znakowego wypisać: ile jest w nim *słów* (poprzez słowo rozumiemy ciąg co najmniej jednego znaku innego niż znak przestankowy, dla uproszczenia przyjmijmy, że liczymy a-z, A-Z i 0-9 jako coś, co składa się na słowa), ile *liter*, ile *cyfr*, oraz wypisać statystykę częstości występowania poszczególnych liter oraz cyfr.

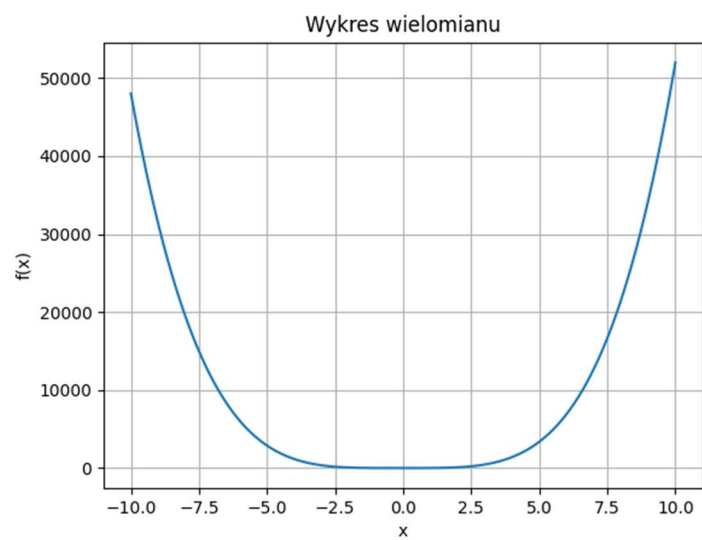
3. Napisać program konwertujący liczby zapisane w systemie rzymskim (wielkimi literami I, V, X, L, C, D, M) na liczby arabskie w zakresie liczb 1-3999, i odwrotnie. Proszę **skontrolować poprawność** danych wejściowy, również w formacie rzymskim. Proszę spróbować napisać najbardziej kompaktowy (krótki) kod.

4. Mamy liczbę naturalną N w zapisie binarnym (czyli składa się tylko z 0 i 1). Binarna przerwa to sekwencja zer otoczonych z lewej i prawej strony 1. Na przykład liczba 9 (decymalnie) binarnie wynosi 1001 i posiada jedną binarną przerwę długości 2. Liczba 529 ma binarną reprezentację 1000010001, zatem ma dwie binarne przerwy, o długości 4 i 3. Liczba 20 ma reprezentację 10100 zawiera zatem jedną binarną przerwę o długości 1. Liczba 15 ma reprezentację 1111, a zatem żadnej binarnej przerwy. Napisz funkcję:

```
def fun(N)
```

która dla podanej liczby naturalnej N (uwaga: liczby w systemie dziesiętnym) zwraca długość jej najdłuższej binarnej przerwy, albo 0, jeśli nie ma ani jednej przerwy. Na przykład, dla  $N = 1041$ , które binarnie jest 10000010001, ma najdłuższą przerwę binarną 5. Należy przyjąć, że argument N może być z przedziału `[1..2147483647]`. Wskazówka: warto skorzystać z operatora przesunięcia bitowego `>>`. Można podejrzeć jak wygląda liczba w zapisie binarnym poprzez rzutowanie `bin(N)`.

5. Python jest językiem, w którym przy użyciu kodu o niewielkiej długości, ale z umiejętnym użyciem bibliotek, da się osiągnąć interesujące wyniki. Wymaga to jednak zapoznania się z możliwościami różnych bibliotek, celem tego zadania jest właśnie podstawowe użycie biblioteki do rysowania (matplotlib) oraz biblioteki numpy. Zadanie: napisać prosty i zwięzły program, który pozwoli na wczytanie wielomianu funkcji  $f(x)$  jako danej wejściowej (łańcuch znakowy) oraz przedział  $x$  (od  $-x_{\min}$ , do  $-x_{\max}$ ). Cel: narysować ten wielomian za pomocą `plt.plot(x_val, y_val)` (gdzie `x_val` i `y_val` to będą, odpowiednio, tablica wygenerowana za pomocą `x_val = np.linspace(x_min, x_max, 200)`, a tablica `y_val` wyliczona z użyciem funkcji `eval` dla wartości z tablicy `x_val`. Przykładowo, jeśli wpiszę na wejściu:  $5x^4 + 2x^3 - x + 6$  i podam -10, 10, to otrzymam:



Proszę, tak jak na rysunku przykładowym widać, dodać **podpis** **osi X i Y**, jakiś **tytuł**, oraz „grid lines”.