Zagadnienia na Test Praktyczny I (ok 7-8 zadań): ogólnie, tematy poruszane na zajęciach 1-4, w szczególności:

- typy danych i dedykowane im metody

- instrukcje warunkowe if/else, pętle for/while, definiowanie funkcji

- listy składane, funkcja anonimowa lambda, funkcje map i filter

- obsługa danych od użytkownika (input)

- praca z plikami tekstowymi zewnętrznymi (open)

- moduł math

Czego nie będzie na pierwszym teście?

- inne moduły niż math

- funkcje rekurencyjne

Zadania powtórkowe (typy zadań):

1. Co pojawi się na ekranie po wywołaniu kodu, np.:
   a) jednolinijkowe

```
x = [3,4,5,6,7,8]
print(x[2])
print(x[2:5])
print(x[1:-2])
print(x[1:5:2])
```

   b) dłuższe

```
y = []
for i in range(5,10,2):
    if i**2 - 1 < 50:
        y.append(i)
print(y)
```

2. Krótki kod do napisania
   a) W jaki sposób odwołać się do wartości 3?

```
x = [[1,[2,[3]]],[4]]
```

   b) Za pomocą **listy składanej** wygeneruj listę 10 elementową składającą się z 1, 11, 111 itd.. aż do 1111111111.
3. Wskaż błędy składniowe w kodzie, np. w

```
[1, 3, 2)
```

4. Zadania programistyczne, np. Zdefiniuj funkcję bezargumentową, która otwiera plik tekstowy „liczby” zawierający pewne liczby całkowite znajdujące się w kolejnych wierszach. Wybierz z tej puli dodatnie liczby, wyznacz średnią arytmetyczną i tę wartość zapisz do innego pliku tekstowego. Dodatkowo funkcja zwraca wartość średniej arytmetycznej.

Zadania programistyczne do samodzielnego przećwiczenia:

Zadanie1: Zdefiniuj funkcję **roznice**, która jako argument przyjmuje listę, a która zwraca nową listę składającą się z różnic pomiędzy kolejnymi elementami wejściowej listy.

Przykładowo roznice([1,2,5,6]) powinna zwrócić listę [1,3,1]

Zadanie2: Zdefiniuj funkcję **dodatnia_lista**, która jako argument przyjmuje bliżej nieokreśloną liczbę list. A która zwraca listę składającą się tylko z dodatnich elementów tych list.

Przykładowo dodatnia_lista([1,2,-3],[5,6,0],[-1,-1,3]) zwróci [1,2,5,6,3]

Zadanie3: Na 3 sposoby wygeneruj listę kwadratów 10 kolejnych liczb naturalnych z wyjątkiem tych podzielnych przez 3. Tzn [1, 4, 16, 25, …, 100]

Zadanie4: Zdefiniuj funkcję **nested_list**, która zwraca listę kolejnych zagnieżdżonych n jedynek

Przykładowe wywołanie nested_list(4) powinno zwrócić [1,[1],[[1]],[[[1]]]]

Zadanie5: Plik tekstowy zawiera 4 sekwencje nukleotydowe – każda w nowej linijce

ATGCTAGAG
TGCGCTTAAAAA
AAAAAAAAAG
CG

Zdefiniuj funkcję **ile_tymin** która jako argument przyjmuje nazwę pliku tekstowego, a która posiada także parametr opcjonalny x będący numerem wiersza (jeżeli użytkownik nie poda tego parametru niech domyślnie będzie to pierwszy wiersz – indeks 0 w Python) i która zwraca liczbę tymin danej sekwencji.