

Podstawy Programowania

Instrukcja do laboratorium 2

1 Zasady ogólne

Ta instrukcja jest przeznaczona na 6 godzin lekcyjnych. Do wykonania i przesłania jest 12 zadań domowych. Ocena za laboratorium zależy od liczby dobrze zrobionych zadań, skala ocen jest pokazana w Tabeli 1.

Tabela 1: Skala ocen.	
Liczba zrobionych zadań domowych	Ocena
6	3.0
7	3.5
8-9	4.0
10-11	4.5
12	5.0

Kod wszystkich wykonanych programów proszę wkleić do jednego pliku, podpisując poszczególne zadania za pomocą komentarzy. Zadanie proszę wklejać po kolei (1, 2, 3...). Plik z zadaniami proszę przesłać na platformie Moodle. Plik musi mieć nazwę `numeralbumu_lab2.py`. **Plik musi być plikiem tekstowym z rozszerzeniem .py.** Wysłanie pracy w nieodpowiedniej formie może skutkować wystawieniem oceny niedostatecznej za te laboratorium.

UWAGA: Termin oddania zadania jest ustawiony w systemie moodle. W przypadku nie oddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia. Zadania oddawane później niż miesiąc po terminie ustawionym na moodle są oddawane i rozliczane w trybie indywidualnym na zajęciach lub po umówieniu się z prowadzącym.

UWAGA: W przypadku wysłania zadania w formie niezgodnej z opisem w instrukcji prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Przykład: wysłanie .zip lub .pdf tam, gdzie był wymagany plik tekstowy z rozszerzeniem .py.

Listing 1: Przykład dobrze sformatowanego pliku z zadaniami.

```
1 # Zadanie 1
2 a = 4
3 b = 2
4 print(a + b)
5
6 # Zadanie 2
7 # Brak
8
9 # Zadanie 3
10 a = 3
11 b = 2
12 print(a ** b)
13
14 # ...
```

2 Zakres tematyczny

- Funkcje, argumenty pozycyjne (positional) i nazwane (keyword).

- https://www.w3schools.com/python/python_functions.asp
- <https://www.pythonstacks.com/blog/post/positional-keyword-and-default-arguments/>
- Python - funkcje wbudowane `min()`, `max()`, `sum()`, `sorted()`.
 - https://www.w3schools.com/python/ref_func_min.asp
 - https://www.w3schools.com/python/ref_func_max.asp
 - https://www.w3schools.com/python/ref_func_sum.asp
 - https://www.w3schools.com/python/ref_func_sorted.asp
- Operatory `in`, `is`, obiekt `None`.
 - <https://www.positronx.io/understand-python-null-and-none-keyword/>
 - https://www.w3schools.com/python/ref_keyword_in.asp
 - https://www.w3schools.com/python/ref_keyword_is.asp
 - https://www.w3schools.com/python/ref_keyword_none.asp
- Sprawdzenie typu zmiennych (`type`, `isinstance`).
 - https://www.w3schools.com/python/ref_func_type.asp
 - https://www.w3schools.com/python/ref_func_isinstance.asp
- Operacje na listach ciąg dalszy.
 - https://www.w3schools.com/python/python_lists_add.asp
 - https://www.w3schools.com/python/python_lists_remove.asp
 - https://www.w3schools.com/python/python_lists_methods.asp
- Zbiór, słownik - operacje.
 - https://www.w3schools.com/python/python_sets.asp - Zalecam przeczytanie całego rozdziału (po lewej stronie przełączenie).
 - https://www.w3schools.com/python/python_dictionaries.asp - Zalecam przeczytanie całego rozdziału (po lewej stronie przełączenie).
- Kopiowanie kolekcji (list) `copy.copy()`.
 - https://www.w3schools.com/python/ref_list_copy.asp
- Listy składane (List comprehension).
 - https://www.w3schools.com/python/python_lists_comprehension.asp

3 Zadania do wykonania

3.1 Zadania do wykonania na zajęciach

Te zadania są robione w trakcie zajęć, nie trzeba ich przysyłać na moodlu.

1. Napisz funkcję która przyjmuje liczbę x jako jedyny argument i zwraca wartość funkcji $f(x) = x^2 + 4x - 3$ dla podanego argumentu x .
Użyteczne wyrażenia: `def`, `return`.
2. Napisz funkcję, która dodaje dwie liczby a i b podane w argumentach. Funkcja ma być zdefiniowana z jednym pozycyjnym (positional) i argumentem nazwanym (keyword/ze zdefiniowaną wartością domyślną). Wartość domyślną dla drugiej liczby proszę ustawić na 2. Następnie, wywołaj tą funkcję na 5 różnych możliwych sposobów tak, żeby została obliczona suma liczb 1 i 2.
Użyteczne wyrażenia: `def`, `return`.

3. Napisz funkcję, która przyjmuje jako argument listę liczb całkowitych, tworzy kopie tej listy i modyfikuje jej kopie w następujący sposób:

- Usuwa wszystkie podzielne przez 3 liczby,
- Wkleja wartość -1 przed każdą liczbą parzystą.

Po dokonaniu modyfikacji zmodyfikowana wersja listy ma być zwrócona z funkcji. Przy tym funkcję należy napisać tak, żeby lista, podana jako argument została niezmieniona. Zademonstruj działanie funkcji na liście liczb losowych z zakresu od 0 do 10.

Listing 2: Przykładowe działanie programu.

```
1 Lista: [2, 3, 1, 5, 6, 3, 2, 4]
2 Wynik: [-1, 2, 1, 5, -1, 2, -1, 4]
```

Użyteczne wyrażenia: `def`, `return`, `copy()`, `del`, `pop()`, `insert()`.

4. Utwórz funkcję, która przyjmuje trzy argumenty n , a , b i zwraca listę o długości n , wypełnioną losowymi liczbami całkowitymi z zakresu od a do b . Argument n ma mieć wartość domyślną 10, argument a wartość domyślną 0 i argument b wartość domyślną 10.

Użyteczne wyrażenia: `def`, `return`, `random.randrange()`.

5. Napisz funkcję, która przyjmując jeden argument n , i wykonuje następujące rzeczy:

- Utworzy listę liczb losowych z zakresu od 0 do 10 o długości n (można użyć funkcje z poprzedniego zadania).
- Wypisze na ekran wylosowaną listę.
- Wypisze na ekran największą liczbę z listy.
- Wypisze indeks pod którym znajduje się najmniejsza liczba w liście.
- Wypisze sumę liczb w liście.
- Wypisze ile razy w liście występuje wartość 5.
- Wypisze zbiór liczb unikalnych występujących w tej liście.

Listing 3: Przykład działania funkcji wywołanej z argumentem $n = 10$.

```
1 [1, 6, 1, 4, 8, 8, 1, 5, 8, 8]
2 8
3 0
4 50
5 1
6 {1, 4, 5, 6, 8}
```

Użyteczne wyrażenia: `print()`, `max()`, `min()`, `index()`, `sum()`, `count()`, `set()`.

6. Napisz funkcję, która przyjmuje dwie listy a i b o dowolnych długościach jako argumenty i zwraca iloczyn kartezjański elementów tych dwóch list w postaci listy krotek (list of tuples).

Listing 4: Przykład działania programu.

```
1 Lista 1: [1, 2, 3]
2 Lista 2: ['a', 'b']
3 Iloczyn kartezjański: [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b'), (3, 'a'), (3, 'b')]
```

Użyteczne wyrażenia: `for` `in`, `def`, `return`, `[]`.

Zabronione w tym zadaniu wyrażenia: `itertools.product()`.

7. Zaznajom się ze składnią list składanych (list comprehension), następnie napisz program, który utworzy używając tej składni trzy listy:

- Lista składająca się z kwadratów liczb naturalnych od 0 do 10.
- Lista składająca się z kwadratów liczb naturalnych od 0 do 20 pod warunkiem że kwadrat tej liczby jest nieparzysty.
- Lista składająca się z na przemian kwadratów i sześciątów liczb naturalnych od 0 do 10. Jeżeli liczba jest parzysta, to zapisujemy jej kwadrat, inaczej sześciąt. *Początek listy: [0, 1, 4, 27, 16, 125....*

Użyteczne wyrażenia: List comprehension.

Zabronione w tym zadaniu wyrażenia: `append()`, `insert()`.

8. Napisz program, który wygeneruje listę składającą się z 20 całkowitych liczb losowych od -100 do 100, a następnie obliczy sumę liczb dodatnich z tej listy, korzystając się ze składni list składanych (list comprehension).

Użyteczne wyrażenia: List comprehension, `random.randrange()`.

9. Napisz funkcję, która przyjmuje jeden argument który może być liczbą albo listą liczb. Jeżeli jako argument została podana liczba, należy zwrócić tą liczbę pomnożoną razy 2. Jeżeli jako argument została podana lista należy utworzyć nową listę, która będzie zawierała elementy pierwotnej listy, pomnożone razy 2. Zadeemonstruj działanie tej funkcji dla liczby i listy losowej.

Użyteczne wyrażenia: List comprehension, `random.randrange()`, `isinstance()`.

10. Napisz funkcję, która zamieni znaki podanego jako argument napisu `s` na liste odpowiadających tym znakom kody ascii. Ta lista ma być zwrócona jako wynik działania tej funkcji. Do napisania funkcji użyj składni list comprehension.

Listing 5: Przykładowe wywołanie funkcji.

```
1 >>> s = 'Ala ma kota'
2 >>> zamien(s)
3 [65, 108, 97, 32, 109, 97, 32, 107, 111, 116, 97]
```

Użyteczne wyrażenia: List comprehension, `ord()`.

11. Napisz funkcję, która przyjmuje dwa napisy (słowa czy zdania), a następnie w kolejnych wierszach wypisuje:
 - zbiór wszystkich znaków unikatowych występujących w obu napisach,
 - zbiór znaków występujących w obu napisach jednocześnie,
 - zbiór znaków występujących w pierwszym napisie, ale nie występujących w drugim.

Listing 6: Przykładowe działanie programu.

```
1 Napis 1: "ala ma kota"
2 Napis 2: "apple"
3 Wynik:
4 {'p', 'o', 'k', 'l', 't', ' ', 'm', 'a', 'e'}
5 {'a', 'l'}
6 {'o', 'k', 't', ' ', 'm'}
```

Użyteczne wyrażenia: `set()`, `intersection()`, `union()`, `difference()`.

12. Napisz funkcję, która będzie posługiwać się predefiniowanym słownikiem do tłumaczenia słów zawierających się w tym słowniku na język polski. Funkcja ma przyjmować napis, będący zbiorem słów w języku angielskim (wystarczy około 5 słów) i zwracać napis składający się z przetłumaczonych słów jeżeli słowo zawiera się w słowniku i oryginałów słów zawartych w nawiasy kwadratowe jeżeli słowo jest nieznane.

Listing 7: Słownik i przykładowe działanie programu.

```
1 Słownik: {
2     'banana': 'banan',
3     'cherry': 'wiśnia',
4     'apple': 'jabłko',
```

```

5 |     'pear': 'grusza',
6 |     'watermelon': 'arbuz'
7 | }
8 | Napis: 'apple banana lemon orange pear'
9 | Wynik: 'jabłko banan [lemon] [orange] grusza'

```

Użyteczne wyrażenia: `split()`, `join()`, `get()`, `if ... is None`.

3.2 Zadania domowe

Zadania przeznaczone do samodzielnego zrobienia w domu. Rozwiązania tych zadań należy przesłać na moodlu.

1. Napisz funkcję która przyjmuje liczbę x jako jedyny argument i zwraca wartość funkcji $f(x) = x^3 - 3x^2 + 8x - 2$ dla podanego argumentu x .

Użyteczne wyrażenia: `def`, `return`.

2. Napisz funkcję, która przyjmuje jako argumenty dwie liczby a i b . Funkcja ma służyć do rozwiązywania równań liniowych typu $ax + b = 0$. Wystarczy że funkcja będzie zwracać (`return`) pierwiastek w przypadku gdy istnieje tylko jedno rozwiązanie równania i wypisywać stosowny komunikat w pozostałych dwóch przypadkach. Funkcja ma być zdefiniowana z jednym pozycyjnym (positional) i argumentem nazwanym (keyword/ze zdefiniowaną wartością domyślną). Wartość domyślną dla argumentu b proszę ustawić na 0. Następnie, wywołaj tą funkcję na 5 różnych możliwych sposobów tak, żeby został obliczony pierwiastek równania $x + 2 = 0$.

Użyteczne wyrażenia: `def`, `return`, `print()`.

3. Napisz funkcję, która przyjmuje jako argument listę liczb całkowitych, tworzy kopie tej listy i modyfikuje jej kopie w następujący sposób:

- Usuwa wszystkie podzielne przez 4 liczby,
- Wkleja wartość -1 przed każdą liczbą nieparzystą.

Po dokonaniu modyfikacji zmodyfikowana wersja listy ma być zwrócona z funkcji. Przy tym funkcję należy napisać tak, żeby lista, podana jako argument została niezmieniona. Zademonstruj działanie funkcji na liście liczb losowych z zakresu od 0 do 10.

Listing 8: Przykładowe działanie programu.

```

1 | Lista: [2, 3, 1, 5, 8, 3, 2, 4]
2 | Wynik: [2, -1, 3, -1, 5, -1, 3, 2]

```

Użyteczne wyrażenia: `def`, `return`, `copy()`, `del`, `pop()`, `insert()`.

4. Utwórz funkcję, która przyjmuje trzy argumenty n , a , b i zwraca listę o długości n , wypełnioną losowymi liczbami całkowitymi z zakresu od a do b . Argument n ma mieć wartość domyślną 10, argument a wartość domyślną 0 i argument b wartość domyślną 10.

Użyteczne wyrażenia: `def`, `return`, `random.randrange()`.

5. Napisz funkcję, która przyjmuje jeden argument n , i wykonuje następujące rzeczy:
 - Utworzy listę liczb losowych z zakresu od 0 do 10 o długości n (można użyć funkcję z poprzedniego zadania).
 - Wypisze na ekran wylosowaną listę.
 - Wypisze na ekran najmniejszą liczbę z listy.
 - Wypisze indeks pod którym znajduje się największa liczba w liście.
 - Wypisze sumę liczb w liście.
 - Wypisze posortowaną listę liczb (przy tym zachowując też wersję nieposortowaną).
 - Wypisze ile razy w liście występuje wartość 3.

- Wypisze zbiór liczb unikalnych występujących w tej liście.

Listing 9: Przykład działania funkcji wywołanej z argumentem $n = 10$.

```
1 [1, 6, 1, 4, 8, 8, 1, 5, 8, 8]
2 8
3 0
4 50
5 1
6 {1, 4, 5, 6, 8}
```

Użyteczne wyrażenia: `print()`, `max()`, `min()`, `index()`, `sum()`, `count()`, `set()`, `sorted()`.

6. Napisz funkcję, która przyjmuje napis o dowolnej długości jako jedyny argument i zwraca listę napisów będących kombinacjami par symboli, z których składa się podany jako argument napis.

Listing 10: Przykład działania programu.

```
1 Napis: 'ABCD'
2 Kombinacje: ['AB', 'AC', 'AD', 'BC', 'BD', 'CD']
```

Użyteczne wyrażenia: `for`, `in`, `def`, `return`, `range()`, `enumerate()`.

Zabronione w tym zadaniu wyrażenia: `itertools.combinations()`.

7. Zaznamoj się ze składnią list składanych (list comprehension), następnie napisz program, który utworzy używając tej składni trzy listy:
 - Lista składająca się z sześciątów liczb naturalnych od 0 do 10.
 - Lista składająca się z sześciątów liczb naturalnych od 0 do 10 pod warunkiem że sześciąt tej liczby jest podzielny przez 3.
 - Lista składająca się z na przemian kwadratów i sześciątów liczb naturalnych od 0 do 10. Jeżeli liczba jest podzielna na 3, to zapisujemy jej sześciąt, inaczej kwadrat. *Początek listy:* $[0, 1, 4, 27, 16, \dots]$

Użyteczne wyrażenia: List comprehension.

Zabronione w tym zadaniu wyrażenia: `append()`, `insert()`.

8. Napisz program, który wygeneruje listę składającą się z 20 całkowitych liczb losowych od -10 do 10, a następnie obliczy sumę kwadratów liczb ujemnych z tej listy, korzystając się ze składni list składanych (list comprehension).

Listing 11: Przykładowe działanie programu (dla uproszczenia pokazany przykład z krótszą listą).

```
1 Lista: [-4, 5, 2, -3, 6]
2 Suma kwadratów liczb ujemnych: 25
```

Użyteczne wyrażenia: List comprehension, `random.randrange()`.

9. Napisz funkcję, która przyjmuje jeden argument który może być liczbą albo listą liczb. Jeżeli jako argument została podana liczba, należy zwrócić tą liczbę podniesioną do kwadratu. Jeżeli jako argument została podana lista należy utworzyć nową listę, która będzie zawierała elementy pierwotnej listy, podniesione do kwadratu. Zademonstruj działanie tej funkcji dla liczby i listy losowej.

Użyteczne wyrażenia: List comprehension, `random.randrange()`, `isinstance()`.

10. Napisz funkcję, która zamieni listę kodów ASCII podawaną jako argument na napis. Ten napis ma być zwrócony jako wynik działania tej funkcji. Do napisania funkcji użyj składni list comprehension. Następnie, wygeneruj listę składającą się z 10 całkowitych liczb losowych z zakresu od 97 (kod ASCII małej litery 'a') do 122 (kod ASCII małej litery 'z') włącznie i użyj poprzednio zaimplementowanej funkcji do zamiany tej listy na losowy ciąg liter.

Listing 12: Przykładowe wywołanie funkcji.

```
1 Przykładowa lista losowa: [122, 108, 107, 118, 103, 107, 102, 111, 104, 115]
2 Otrzymany losowy ciąg liter: 'zlkvgkfohs'
```

Użyteczne wyrażenia: List comprehension, chr(), random.randrange(), join().

11. Napisz funkcję, która przyjmuje dwie listy składających się z liczb losowych a następnie w kolejnych wierszach wypisuje:

- zbiór wszystkich liczb unikatowych występujących w obu listach,
- zbiór liczb występujących w obu listach jednocześnie,
- zbiór liczb występujących w pierwszej liście, ale nie występującej w drugiej.

Następnie należy wygenerować dwie listy składające się z 10 całkowitych liczb losowych z zakresu od 0 do 10 i zademonstrować działanie zaimplementowanej funkcji.

Listing 13: Przykładowe działanie programu (dla uproszczenia są podane krótsze listy).

```
1 Lista 1: [1, 2, 1, 3, 4]
2 Lista 2: [2, 5, 5, 2]
3 Wynik:
4 {1, 2, 3, 4, 5}
5 {2}
6 {1, 3, 4}
```

Użyteczne wyrażenia: set(), intersection(), union(), difference().

12. Napisz program, który będzie pobierał od użytkownika słowa w języku angielskim i tłumaczył na język polski za pomocą predefiniowanego słownika. Jeżeli wprowadzone słowo nie zawiera się w słowniku należy zapytać użytkownika czy on chce dodać słowo wraz z tłumaczeniem do słownika. Jeżeli użytkownik odpowie tak, to należy zapytać się o tłumaczenie tego słowa i dodać parę nieznanne słowo - tłumaczenie do słownika. Program ma działać w pętli do momentu wprowadzenia przez użytkownika pustego wierszu (po prostu wciśnięcie Enter).

Listing 14: Słownik i przykładowe działanie programu.

```
1 Słownik: {
2     'banana': 'banan',
3     'cherry': 'wiśnia',
4     'apple': 'jabłko',
5     'pear': 'grusza',
6     'watermelon': 'arbuz'
7 }
8
9 Wprowadź słowo: cherry
10 wiśnia
11 Wprowadź słowo: lemon
12 Słowo jest nieznanne. Chcesz go dodać? [tak/nie]: tak
13 Wpisz tłumaczenie tego słowa: cytryna
14 Wprowadź słowo: lemon
15 cytryna
16 Wprowadź słowo:
17 Koniec programu.
```

Użyteczne wyrażenia: while, input(), break, if ... is None.