

Zadania łatwe (1 pkt)

Z1:

Opracuj program, który będzie prosił użytkownika o podanie liczby sekund i będzie działał następująco.

- Jedna minuta zawiera 60 sekund. Jeżeli liczba sekund, którą podał użytkownik, jest większa niż 60, program powinien wyświetlić liczbę minut i sekund odpowiadającą danej liczbie sekund.
- Jedna godzina zawiera 3600 sekund. Jeżeli liczba sekund, którą podał użytkownik, jest większa niż 3600, program powinien wyświetlić liczbę godzin, minut i sekund odpowiadającą danej liczbie sekund.
- Jeden dzień to 86400 sekund. Jeżeli liczba sekund, którą wprowadził użytkownik, jest większa niż 86400, program powinien wyświetlić liczbę dni, godzin, minut i sekund odpowiadającą danej liczbie sekund.

Z2:

Napisz skrypt, którego zadaniem będzie zakodowanie wprowadzonego przez użytkownika ciągu tekstowego z użyciem szyfru Cezara o zadanym kluczu. W tej metodzie szyfrowania, każda litera tekstu jawnego zastępowana jest literą znajdującą się określoną liczbę miejsc (klucz) dalej w alfabecie. Opcja (+1 pkt.): kodowany/zakodowany tekst jest odczytywany/zapisywany z/do pliku.

Z3:

Napisz skrypt odczytujący z pliku tekstowego zapisane w kolumnie wartości temperatury w stopniach kelwina. Skrypt ma przeliczyć odczytane wartości na stopnie Celsjusza i zapisać do pliku *Celsjusz.txt*, oraz wyświetlić na konsoli wartości maksymalną, minimalną i średnią temperatur.

Z4:

Luty ma standardowo 28 dni, natomiast w roku przestępnym 29. Opracuj program proszący użytkownika o podanie roku. Zadaniem tego programu będzie wyświetlenie liczby dni lutego w danym roku. Do identyfikacji roku przestępnego wykorzystaj następujące kryteria.

1. Ustal, czy rok jest podzielny przez 100. Jeżeli tak, będzie rokiem przestępnym, gdy jest podzielny również przez 400. Dlatego też rok 2000 jest przestępny, natomiast 2100 już nie.
2. Jeżeli rok nie jest podzielny przez 100, będzie przestępny tylko wtedy, gdy jest podzielny przez 4. Dlatego też rok 2008 jest przestępny, natomiast 2009 już nie.

Oto zapis przykładowej sesji działania tego programu:

Podaj rok: 2008 (Enter) W 2008 roku luty ma 29 dni

Z5:

Napisz skrypt, który odczyta informacje zapisane w pliku *carbon.txt* (kolejne kolumny: nazwa kraju, emisja CO₂ w mln ton, emisja w przeliczeniu na jednego mieszkańca). Przedstaw dane na wykresie słupkowy (*bar* w *matplotlib*). Opcja (+1 pkt.), skrypt umożliwi przedstawienie wybranej liczby krajów o największej emisji CO₂ łącznie bądź w przeliczeniu na jednego mieszkańca.

Z6:

Jeżeli średnio aktywna osoba codziennie ograniczy liczbę kalorii o 500, będzie mogła schudnąć 2 kg miesięcznie. Utwórz program pozwalający użytkownikowi na podanie wagi początkowej, a następnie generujący tabelę pokazującą oczekiwaną wagę na końcu każdego miesiąca w najbliższym półroczu, o ile uda jej się utrzymać dietę.

Z7:

Opracuj program przewidujący przybliżoną wielkość populacji pewnych organizmów. Aplikacja powinna umożliwić użytkownikowi podanie liczby początkowej organizmów, wyrażonego procentowo dziennego przyrostu populacji i liczby dni, przez które organizmy pozostaną, aby mogły się mnożyć. Jeśli np. użytkownik poda następujące dane:

Początkowa liczba organizmów: 2

Dzienny przyrost: 30%

Liczba dni: 10

wówczas program powinien wyświetlić następującą tabelę:

Dzień	Populacja
1	2
2	2,6
3	3,38
4	4,394
5	5,7122
6	7,42586
7	9,653619
8	12,5497
9	16,31462
10	21,209

Z8

Opracuj program, który będzie generował proste zadania matematyczne. Program powinien wyświetlić dwie liczby losowe, które uczeń będzie musiał do siebie dodać: $247 + 129$. Program powinien poprosić ucznia o podanie wyniku dodawania i jeżeli odpowiedź jest prawidłowa, wyświetlić komunikat z gratulacjami, a jeśli odpowiedź jest błędna, wyświetlić komunikat z prawidłowym rozwiązaniem.

Z9

Opracuj program proszący użytkownika o podanie serii 20 liczb. Program powinien je przechowywać na liście i wyświetlać dane, takie jak:

- najmniejsza liczba na liście;
- największa liczba na liście;
- suma wszystkich liczb na liście;
- średnia wielkość liczb na liście.

Z10

Napisz skrypt symulujący totolotek, czyli generujący 6 liczb losowych bez powtórzeń, z przedziału od 1 do 49

Zadania trudne (2 pkt)

Z1:

Stwórz program obliczający intensywność promieniowania X po przejściu przez warstwy różnych tkanek. Skorzystaj z równania: $I = I_0 e^{-\mu x}$, gdzie I_0 to początkowa intensywność, μ to liniowy współczynnik pochłaniania (różny dla kości od -190 do -130, mięśni -29 do +150, tłuszczu, >+150), a x to grubość warstwy. Zilustruj wyniki na wykresie dla wspomnianych trzech tkanek, stałe x .

Z2:

Dana jest lista liczb całkowitych **nums** i liczba całkowita **target**. Napisz funkcję, która zwróci wartość indeksów dwóch liczb z listy **nums** takich, że suma tych liczb jest równa **target**.

Przykładowo: `nums = [2,7,11,15]`, `target = 9`

output: `[0,1]`

Z3:

Napisz skrypt wyznaczający wartość liczby π metodą Monte Carlo.

Opis algorytmu:

- Wpisz koło o promieniu r w kwadrat o boku $2r$
- Losowo wygeneruj punkty i umieść je w kwadracie
- Wyznacz liczbę punktów, które znajdują się jednocześnie w kwadracie i w kole
- Niech promień r będzie wyznaczony przez stosunek liczby punktów znajdujących się w kole do liczby punktów znajdujących się w kwadracie
- $\pi = 4.0 \cdot r$

Z4:

Wygeneruj listę liczb odpowiadającą czasowi rozpadu (t) od 0 do 100 sekund z krokiem 10 sekund. Oblicz i wypisz ilość pierwiastka promieniotwórczego pozostałego po czasie t (np. $N(t) = N_0 e^{-\lambda t}$, przyjmij $\lambda = 0,1$). Zaproponuj funkcję obliczającą ilość pierwiastka, przyjmującą jako argumenty wartość początkową N oraz wektor czasu. Wykreśl zależność czasową ilości pierwiastka.

Z5:

Utwórz program, który będzie prosił użytkownika o podanie ocen z danego przedmiotu. Program powinien wyświetlić listę ocen i ocenę średnią. W programie przygotuj następujące funkcje:

- `calc_average()` — powinna przyjmować jako argument listę ocen i zwracać średni wynik;
- `determine_grade()` — powinna przyjmować jako argument średnią ocen ocenę odpowiadającą temu wynikowi określoną na podstawie regulaminu studiów.

Z6:

Opracuj program proszący użytkownika o podanie imienia i nazwiska oraz opisującego go zdania. Oto zapis przykładowej sesji pracy z programem:

Podaj imię: Julia Talar (Enter)

Przedstaw się: Jestem studentką UG, członkiem klubu jazzowego, a po zakończeniu nauki chciałabym zająć się tworzeniem aplikacji w Python. (Enter)

Gdy użytkownik wpisze wymagane informacje, program powinien wygenerować

plik HTML zawierający podane dane. Oto przykładowy dokument HTML

wygenerowany na podstawie wcześniej podanych danych:

```
<html>
<head>
</head>
<body>
  <center>
    <h1>Julia Talar</h1>
  </center>
  <hr />
  Jestem studentką informatyki, członkiem klubu jazzowego, a po
  zakończeniu nauki chciałabym zająć się tworzeniem aplikacji mobilnych.
  <hr />
</body>
</html>
```

Z7

Napisz program, który tworzy w systemie bezpieczne hasło. Powinno ono spełniać następujące wymagania:

- musi składać się przynajmniej z siedmiu znaków;
- musi zawierać przynajmniej jedną dużą literę;
- musi zawierać przynajmniej jedną małą literę;
- musi zawierać przynajmniej jedną cyfrę.

Pomocne mogą być predefiniowane zestawy znaków z modułu *string*: *ascii_letters*, *ascii_lowercase*, *ascii_uppercase*, *ascii_digits*

Z8

Twoim zadaniem jest opracowanie programu oceniającego egzamin pisemny studentów. Egzamin składa się z 20 pytań jednokrotnego wyboru. Oto prawidłowe odpowiedzi na pytania:

1. A	5. B	11. C	16. C
2. C	7. C	12. A	17. B
3. A	8. A	13. D	18. B
4. A	9. D	14. A	19. A
5. D	10. A	15. A	20. D

Program powinien przechowywać prawidłowe odpowiedzi na liście. Po odczytaniu z pliku tekstowego odpowiedzi udzielonych przez kandydata na kierowcę dane te powinny zostać umieszczone na drugiej liście. (Do przetestowania aplikacji samodzielnie utwórz plik z odpowiedziami na pytania testu). Po odczytaniu odpowiedzi osoby egzaminowanej program powinien wyświetlić komunikat informujący o wyniku egzaminu. (Zaliczenie wymaga prawidłowej odpowiedzi na 11 z 20 pytań). Wygenerowane przez program dane wyjściowe powinny zawierać całkowitą liczbę prawidłowych i nieprawidłowych odpowiedzi, a także numery pytań, na które kandydat udzielił błędnych odpowiedzi.