

## Lista zadań nr 2

Funkcje, ciągi tekstowe, krotki, listy.

**Zadanie 1** Zdefiniuj funkcję `square()`, która obliczy wartość trójmianu  $ax^2 + bx + c$  w punkcie o wartości  $x$ . Argumentami funkcji powinny być kolejne wartości współczynników  $a, b, c$  oraz wartość  $x$  jako czwarty argument. Sprawdź, czy wartość trójmianu  $x^2 + 3x + 1$  dla  $x = 2$  jest równa 11.

**Zadanie 2** Zdefiniuj funkcję konwertującą temperaturę w skali Celsjusza na skalę Fahrenheita - funkcja powinna przyjmować argument w postaci temperatury w skali Celsjusza a zwracać wartość wyrażoną w stopniach Fahrenheita. Wykorzystaj zależność:

$$T_{Fahrenheit} = \frac{9}{5}T_{Celsjusz} + 32.$$

Następnie napisz program, który będzie konwertował podawane przez użytkownika temperaturę dopóki będzie ona nie mniejsza niż zero bezwzględne (przyjmij, że zero bezwzględne to  $-273.15^{\circ}\text{C}$ )

**Zadanie 3** Zdefiniuj funkcję `f()`, która przyjmuje jako argumenty dwie liczby i nie zwraca wartości. Zadaniem funkcji jest wyświetlenie komunikatu na wzór przykładu:

`liczba 7 podzielona przez 3 daje 2 i reszta 1`

gdy wywołamy `f(7,3)`. Przetestuj funkcję w prostym programie.

**Zadanie 4** Zdefiniuj funkcję, która przyjmuje cztery argumenty (dowolne liczby) i zwraca średnią arytmetyczną trzech największych z nich. W definicji możesz skorzystać tylko z funkcji wbudowanych `min()`, `max()` i operatorów arytmetycznych. Nie możesz korzystać z pętli i instrukcji warunkowej. Przetestuj funkcję w prostym programie.

**Zadanie 5** Utwórz funkcję o nazwie `order_Tshirt`, przyjmującą jako argumenty: wielkość koszulki ( S, M, L, XL); kolor; oraz tekst, który ma być na niej nadrukowany. Funkcja powinna wyświetlać zdanie zawierające informacje dotyczące zamówionej koszulki: jej rozmiar, kolor i tekst nadruku. Napisz program w którym "zamawiasz" dwie koszulki. W trakcie pierwszego wywołania funkcji do przygotowania koszulki zastosuj argumenty pozycyjne. Natomiast w trakcie drugiego wywołania użyj argumentów w postaci słów kluczowych.

**Zadanie 6** Zmodyfikuj funkcję `order_Tshirt` z zadania 5 tak, aby domyślnie były zamawiane czarne koszulki w rozmiarze M z nadrukowanym tekstem "Uwielbiam Pythona". Zamów koszulkę czarną w rozmiarze L i domyślnym nadrukiem, oraz czerwoną koszulkę o domyślnym rozmiarze i nadruku.

**Zadanie 7** Napisz funkcję o nazwie `collatz`, która pobiera tylko jeden argument o nazwie `number`. Jeżeli argument funkcji jest parzysty, funkcja powinna wyświetlić i zwrócić wartość `number // 2`. Jeżeli argument funkcji będzie nieparzysty wtedy funkcja `collatz` powinna wyświetlić i zwrócić wartość `3 * number + 1`. Napisz program, w którym użytkownik podaje pewną liczbę całkowitą `n`, a program oblicza wartość funkcji `collatz(n)` oraz wartości tej funkcji z kolejnych wartości zwracanych przez uprzednie wywołania funkcji `collatz`, aż do momentu, gdy wartością zwróconą będzie liczba 1. To zaskakujące, ale dla dowolnej liczby całkowitej `n`, wcześniej czy później otrzymana sekwencja zwracanych liczb zakończy się jedynką. Opisane zagadnienie nosi nazwę *problemu Collatza* i do dzisiaj nie jest to problem rozwiązany przez matematyków tzn. nie ma dowodu, że rzeczywiście każda sekwencja liczb będzie kończyć się jedynką dla dowolnej liczby całkowitej `n`, będącej liczbą początkową sekwencji.

**Zadanie 8** Napisz program, który symuluje grę w kości. Aby wygrać, w pierwszym rzucie gracz musi wyrzucić na dwóch kościach sumę oczek równą 7 albo 11. Jeżeli wyrzuci sumę 2, 3 lub 12, przegrywa. Każdy inny wynik to tzw. "punkt" zezwalający na kontynuację gry. W kolejnych rzutach gracz odnosi zwycięstwo, jeżeli ponownie wyrzuci "punkt", a przegrywa przez wyrzucenie sumy 7. Na końcu każdej gry program ma zapytać użytkownika, czy gra jeszcze raz. Gdy użytkownik zdecyduje o zakończeniu rozgrywki, program ma wypisać liczbę przegranych i wygranych gier i zakończyć działanie. Program powinien wykorzystywać następujące funkcje: funkcja `throw()` – zwracająca wyniki rzut dwoma kośćmi, funkcja `single_game()` – powinna obsługiwać pojedynczą grę poprzez wywołanie funkcji `throw()` dla określenia wyników kolejnych rzutów, a także wyświetlać przebieg gry i zwracać `True` gdy gra zakończy się wygraną gracza lub `False` gdy gracz przegra, funkcja `get_answer()` powinna pobierać prawidłową odpowiedź (tak lub nie) i ją zwracać, funkcja `game()` powinna realizować całą rozgrywkę wywołując odpowiednio funkcje `single_game()` i `get_answer()` oraz zliczać liczbę wygranych i przegranych gracza.

**Zadanie 9** Napisz program, który poprosi użytkownika o podanie dowolnego ciągu tekstowego, a następnie wypisze ten ciąg w odwrotnej kolejności.

**Zadanie 10** Napisz funkcję `is_palindrome()`, która zwraca wartość `true` lub `false` w zależności od tego czy argument funkcji (ciąg tekstowy) jest palindromem<sup>1</sup> czy nie.

**Zadanie 11** Napisz program, który poprosi użytkownika o podanie ciągu tekstowego, a następnie wyświetli ten ciąg bez samogłosek. Wykorzystaj pętle `for`.

---

<sup>1</sup>*Palindrom* to słowo, wyrażenie bądź zdanie, które czytane od lewej do prawej brzmi tak samo. Nazwa wywodzi się od greckiego słowa *palindromos*, które znaczy "biegnąc na powrót". Palindromami są np. słowo "potop" czy wyrażenie "oko w oko".

**Zadanie 12** Napisz program, który zliczy ile razy dana litera pojawia się w podanym przez użytkownika ciągu tekstowym. Program powinien wypisać informacje o liczbie wystąpień każdej litery alfabetu. Program można napisać wykorzystując listy lub, jeszcze prościej, wykorzystując słowniki. Tu wykorzystaj metody ciągów tekstowych, a wyników nigdzie nie musisz zapamiętywać - wystarczy je wyświetlić.

**Zadanie 13** Napisz program, w którym zdefiniowane będą dwie krotki zawierające symbole pierwiastków chemicznych. Pierwsza krotka powinna zawierać symbole metali alkalicznych: *Li*, *Na*, *K*, a druga – symbole fluorowców: *F*, *Cl*, *Br*.

Program powinien generować i wypisywać wzory związków chemicznych powstałych z połączenia każdego pierwiastka z pierwszej krotki z każdym pierwiastkiem z drugiej krotki (np. *LiF*, *LiCl*, itp.).

Do realizacji zadania użyj dwóch zagnieżdżonych pętli `for`.

Powstałe związki chemiczne są poprawne i należą do grupy halogenków, takich jak fluorki, chlorki i bromki.

**Zadanie 14** Napisz grę „Przestawione litery”. Gra polega na tym, że komputer wybiera losowo słowo z określonej grupy słów, a następnie przestawia jego litery w losowej kolejności, tworząc anagram.

Zadaniem gracza jest odgadnięcie oryginalnego słowa na podstawie pomieszanego ciągu liter.

Program powinien składać się z następujących funkcji:

- `choose_word(word_list)` – wybiera losowe słowo z podanej listy;
- `shuffle_letters(word)` – generuje anagram poprzez losowe przemieszanie liter;
- `play_game()` – przeprowadza rozgrywkę, wyświetlając anagram i pobierając odpowiedź od gracza.

**Zadanie 15** Napisz grę „Zgadnij, jakie to słowo”. Gra polega na tym, że komputer losowo wybiera słowo z określonej grupy słów i informuje gracza, ile liter zawiera to słowo.

Gracz ma pięć prób na zadanie komputerowi pytań o to, czy dana litera znajduje się w wybranym słowie.

Komputer może odpowiadać jedynie „TAK” lub „NIE”. Po wykorzystaniu wszystkich prób gracz może spróbować odgadnąć całe słowo.

Program powinien zawierać następujące funkcje:

- `choose_word(word_list)` – wybiera losowe słowo z podanej listy;
- `check_letter(word, letter)` – sprawdza, czy dana litera znajduje się w słowie, i zwraca odpowiedź „TAK” lub „NIE”;

- `play_game()` – przeprowadza pełną rozgrywkę, obsługując pytania gracza i końcowe odgadywanie słowa.

**Zadanie 16** Utwórz listę z dowolnymi napisami. Następnie wygeneruj na jej podstawie listę, która zawiera pierwsze znaki napisów zawartych w Twojej liście.

**Zadanie 17** Sprawdź, jaka jest różnica między instrukcją `L.insert(2, -1)` a przypisaniem `L[2] = -1` dla dowolnej listy `L`.

Metoda `append()` pozwala dodawać elementy na koniec listy. W jaki sposób można dodawać elementy zarówno na początek, jak i na koniec listy, korzystając z metody `insert()`?

**Uwaga:** Twoje rozwiązanie powinno działać dla listy o dowolnej długości.

**Zadanie 18** Napisz program, który utworzy listę kwadratów liczb całkowitych od 0 do 9. Wykorzystaj pętlę `for` oraz metodę `append()`.

Następnie zmodyfikuj program, definiując funkcję, która przyjmie jako argument listę liczb, a następnie zwróci listę ich kwadratów.

**Przykład:** Dla listy `[-1, 2, -6, 9]` funkcja powinna zwrócić `[1, 4, 36, 81]`.

**Zadanie 19** Napisz program do przechowywania pięciu najlepszych wyników gracza.

Program powinien wyświetlać menu umożliwiające wykonywanie operacji na liście wyników, takich jak:

- dodawanie nowego wyniku;
- usuwanie wyniku;
- sortowanie wyników;
- wyświetlanie listy wyników;
- zakończenie pracy z programem.

Na początku lista wyników powinna być pusta. W trakcie działania programu może przechowywać maksymalnie 5 najlepszych wyników. Jeśli zostanie dodany 6. wynik, to lista powinna automatycznie usunąć najniższy wynik, tak aby nadal zawierała tylko najlepsze osiągnięcia.

**Zadanie 20** Napisz program „Generator obelg”, który w wyrafinowany sposób będzie obrażał osobę wskazaną przez użytkownika.

Program powinien poprosić użytkownika o podanie imienia oraz wieku „ofiary”.

W programie powinny być zadeklarowane dwie listy:

- lista przymiotników;
- lista rzeczowników.

Na początku każda z list powinna zawierać przynajmniej dwa elementy.

Program powinien umożliwiać:

- dodawanie i usuwanie przymiotników i rzeczowników (zawsze muszą pozostać przynajmniej dwa słowa);
- generowanie losowej obelgi;
- zakończenie działania programu.

Dodatkowo, w zależności od wieku osoby, program powinien dobierać dodatkowy przymiotnik, który dopasuje obelgę np.: „nieopierzony”, „stary”, „zgrzybiały” itp.

**Zadanie 21** Napisz grę „Wisielec”. Komputer losowo wybiera ukryte słowo, a zadaniem gracza jest stopniowe odgadnięcie go, podając litery jedna po drugiej.

**Zasady gry:**

- Gracz wprowadza pojedyncze litery.
- Jeśli podana litera znajduje się w słowie, jest ujawniana na odpowiednich pozycjach.
- Jeśli litera nie występuje w słowie, komputer wyświetla kolejne etapy rysunku wisielca.
- Gracz wygrywa, jeśli odgadnie całe słowo przed ukończeniem rysunku wisielca.
- Jeśli rysunek zostanie ukończony, gracz przegrywa.