

## LISTA ZADAŃ 1

### ZADANIA DO ROZWIĄZANIA

1. Napisz program obliczający pole trójkąta (wzór Herona).
  - a. Długości boków trójkąta mają być podawane przez użytkownika jako argumenty wywołania programu. Zwróć uwagę na typ podawanych argumentów.
  - b. Wynik obliczeń ma być wypisywany na standardowym wyjściu z dokładnością do trzech miejsc po przecinku.
2. Napisz program obliczający elementy ciągu Fibonacciego.
  - a. Niech funkcja o nazwie *fibonacci* o argumentach wejściowych  $x$ ,  $y$ ,  $n$ , oblicza i zwraca jako argument wyjściowy ciąg  $n$  liczb Fibonacciego począwszy od liczb  $x$  i  $y$ .
  - b. Niech program wywołuje funkcję *fibonacci* z argumentami podanym przez użytkownika jako argumenty wywołania programu i wypisuje wynik zwracany przez funkcję na standardowym wyjściu. Zwróć uwagę na sposób formatowania wyniku.
3. Napisz program obliczający elementy ciągu geometrycznego określonego wzorem  $a[n+1] = a[n]*q$ .
  - a. Zdefiniuj funkcję, która dla zadanych jako argumenty wejściowe wyrazu początkowego  $a[0]$ , ilorazu  $q$  oraz liczby obliczanych początkowych elementów ciągu  $n$  zwraca tablicę elementów.
  - b. Niech program wywołuje funkcję z pkt. a dla przykładowych danych wejściowych i wypisuje wynik na standardowym wyjściu. Zwróć uwagę na sposób formatowania wyniku.
4. Napisz program obliczający tabliczkę mnożenia.
  - a. Do obliczania tabliczki mnożenia liczb od 1 do  $n$ , gdzie  $n$  jest dodatnią liczbą całkowitą, ma służyć funkcja przyjmująca  $n$  jako argument wejściowy i zwracająca dwuwymiarową tablicę z tabliczką mnożenia jako argument wyjściowy.
  - b. Liczba  $n$  ma być podawana przez użytkownika jako argument wywołania programu. Zwróć uwagę na typ i wartość podawanego argumentu.
  - c. Obliczona tabliczka mnożenia ma być wypisywana na standardowym wyjściu. Zwróć uwagę na sposób formatowania wyniku.
5. Napisz program zliczający samogłoski w podanych wyrazach.
  - a. Wyrazy są podawane przez użytkownika na standardowym wejściu, aż do podania wartości kończącej działanie programu (np.  $q$ , od ang. *quit*).
  - b. Zliczane są samogłoski współczesnego alfabetu łacińskiego: a, e, i, o, u, y (obsługa polskich znaków ą, ę, ó nie jest wymagana).
  - c. Podany wyraz wraz z obliczoną liczbą samogłosek jest zapisywany do pliku, którego nazwa została podana jako argument wywołania programu, w formacie:  
(wyraz) (tabulator) (liczba\_samogłosek)  
np. *kot* 1
  - d. Program zawiera obsługę sytuacji, gdy wyraz został znajduje się już w pliku.
6. Dane są dwa pliki tekstowe, w których znajdują się wyrazy oddzielone białymi znakami (np. spacjami, tabulatorami lub znakami przejścia do nowej linii). Napisz program, który:
  - a. przyjmuje jako argument wywołania nazwy dwóch plików tekstowych,
  - b. znajduje wyrazy, które występują w obu plikach,
  - c. wypisuje takie wyrazy na standardowym wyjściu w porządku alfabetycznym.

Małe i wielkie litery powinny być traktowane jako nierozróżnialne. Polskie znaki nie muszą być obsługiwane. Cyfry i znaki interpunkcyjne nie muszą być obsługiwane.

Dla każdego programu oraz nietrywialnej funkcji:

- napisz komentarz dokumentujący zgodny ze standardem języka,
- zaprojektuj oraz przeprowadź kilka testów sprawdzających poprawność.

## UWAGI DOTYCZĄCE ROZWIĄZAŃ

Podczas rozwiązywania zadań zwróć uwagę na następujące kwestie podlegające ocenie:

1. Ile argumentów powinien przyjmować program/funkcja? Czy liczba argumentów podanych przez użytkownika jest sprawdzana? Czy program/funkcja wyświetla odpowiedni (i zrozumiały dla użytkownika) komunikat o błędzie w przypadku podania nieprawidłowej liczby argumentów (za mało/za dużo)?
2. Jakie typy danych powinien przyjmować program/funkcja (np. int, double, string)? Czy poprawność typów jest sprawdzana? Czy program/funkcja wyświetla odpowiedni (i zrozumiały dla użytkownika) komunikat o błędzie w przypadku podania argumentu nieprawidłowego typu?
3. Jakie wartości powinien przyjmować program/funkcja (np. dodatnie/ujemne, całkowite/zmiennoprzecinkowe)? Czy poprawność wartości jest sprawdzana? Czy program/funkcja wyświetla odpowiedni (i zrozumiały dla użytkownika) komunikat o błędzie w przypadku podania nieprawidłowej wartości?
4. Czy typ zwracanych danych jest zgodny z założeniami programu/funkcji (np. int, string, tablica)? Czy wynik wypisywany na standardowy wyjściu/zapisywany do pliku jest prawidłowo (i czytelnie dla użytkownika) sformatowany?
5. Czy program/funkcja zapisujący dane do pliku/odczytujący dane z pliku sprawdza poprawność nazwy i typu pliku? Czy program/funkcja pozwala na podanie zarówno względnej, jak bezwzględnej ścieżki do pliku? Czy program/funkcja wyświetla odpowiedni (i zrozumiały dla użytkownika) komunikat o błędzie w przypadku podania nieprawidłowych wartości (np. nieistniejącej ścieżki)?
6. Czy sprawdzanie typów, wartości i liczby podawanych argumentów znajduje się w odpowiednim miejscu programu/funkcji (np. czy liczba argumentów programu jest sprawdzana przed pierwszą próbą odwołania się do tych argumentów)?
7. Czy w programach wykorzystujących funkcje podział funkcjonalności pomiędzy funkcję a nadrzędny program jest zgodny z założeniami zadania (np. czy funkcja powinna wykonywać obliczenia i wypisywać wynik na standardowym wyjściu, czy tylko wykonywać obliczenia)?
8. Czy w programach wykorzystujących argumenty wywołania programu i standardowe wejście jest zachowany podział pomiędzy te opcje zgodny z założeniami zadania (np. czy jeden parametr jest podawany jako argument wywołania, a drugi ze standardowego wejścia)?

**TERMIN ODDANIA ROZWIĄZAŃ: 24/03/2020**

Agnieszka Kazimierska  
Witold Dyrka