

# Programowanie strukturalne (2023)

## - Kolokwium 1 - Zestaw A01

Rozwiązania mają być umieszczone zgodnie ze specyfikacją:

- Zadania powinny być umieszczone w archiwum .zip na udostępnionym pendrive.
- Nazwa archiwum powinna być wg schematu NUMERZESTAWU\_NUMERALBUMU.zip gdzie numer zestawu znajduje się na górze kartki z poleceniami. np. A23\_123456.zip
- We wnętrzu archiwum powinny znajdować się tylko same kody w języku C, pliki powinny posiadać dokładnie nazwy (z uwzględnieniem wielkości znaków): zad1.c, zad2.c, zad3.c, zad4.c, zad5.c.
- Maksymalna waga archiwum 10 MB.
- Archiwum powinno być bez hasła.

1. W folderze DebugXY (XY - losowe znaki) na pendrive znajduje się projekt z kodem w języku C. W pliku main.c w niektórych liniijkach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych linijek czy zaburzenie struktury kodu oznacza zero punktów za polecenie.

*Punktacja: 5 pkt.*

2. Napisz funkcję, której argumentem jest liczba całkowita  $n$ . Funkcja ma zwrócić sumę cyfr liczby  $n$ . Stwórz przypadek testowy.

Przykład: dla liczny 34 suma cyfr to 7. Dla liczby -67 suma cyfr to 13.

*Punktacja: 7 pkt.*

3. Napisz funkcję `sum_of_squares`, która ma dwa argumenty. Pierwszym argumentem jest wskaźnik `num1` na stałą wartość typu `int`, a drugim argumentem jest stały wskaźnik `num2` na zmienną typu `int`. Funkcja `sum_of_squares` ma zwrócić liczbę całkowitą zawierającą sumę kwadratów wartości wskazywanej przez pierwszy i drugi wskaźnik. Stwórz przypadek testowy dla funkcji.

*Punktacja: 8 pkt.*

4. Napisz rekurencyjną funkcję `sequence_value`, która dla otrzymanej w argumencie nieujemnej liczby całkowitej  $n$  zwraca wartość elementu o indeksie  $n$  ciągu zdefiniowanego w następujący sposób:

$$\begin{aligned}b_0 &= 1 \\b_{3n} &= b_n + 2, n > 0 \\b_{3n+1} &= b_n + 1, n \geq 0 \\b_{3n+2} &= b_n - 1, n \geq 0.\end{aligned}$$

Stwórz dwa przypadki testowe.

*Punktacja: 10 pkt.*

5. Napisz funkcję `sum_odd_indices`, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `int`. Funkcja ma zwrócić sumę elementów znajdujących się na nieparzystych indeksach. Stwórz przypadek testowy.

*Punktacja: 10 pkt - gdy rozwiązanie w całości oparte na wskaźnikach. 7 pkt - gdy rozwiązanie bazuje na notacji tablicowej (przez nawiasy kwadratowe). W przypadku rozwiązania mieszanego, maksymalna liczba punktów może być zmienna.*