

# Programowanie strukturalne (2023)

## - Kolokwium 1 - Zestaw A07

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 liniach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych linii czy zaburzenie struktury kodu oznacza zero punktów za polecenie.

*Punktacja: 5 pkt.*

2. Napisz funkcję, która obliczy przybliżoną wartość liczby  $\pi$  na podstawie sumy  $n$  pierwszych wyrazów szeregu Leibniza:

$$\pi = 4 \sum_{k=0}^n \frac{(-1)^k}{2k+1} = 4 \cdot \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right)$$

Funkcja przyjmuje jako argument dodatnią liczbę naturalną  $n$  i zwraca wartość przybliżonej liczby  $\pi$ . Stwórz przypadek testowy.

*Punktacja: 7 pkt.*

3. Napisz funkcję `absolute_value`, która przyjmuje jako argument wskaźnik na zmienną typu `int`. Funkcja powinna obliczyć wartość bezwzględną zmiennej wskazywanej przez wskaźnik i zaktualizować wartość wskazywaną przez wskaźnik. Stwórz przypadek testowy w `main` tak, aby wyświetlić na konsoli wartość zaktualizowanej zmiennej.

*Punktacja: 8 pkt.*

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

|       |   |    |   |    |     |
|-------|---|----|---|----|-----|
| $n$   | 0 | 1  | 2 | 3  | ... |
| $a_n$ | 3 | -4 | 5 | -6 | ... |

Stwórz dwa przypadki testowe. *Punktacja: 10 pkt.*

5. Napisz funkcję `count_elements_below_average`, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `int`. Funkcja ma zwrócić liczbę elementów tablicy, które są mniejsze od średniej wartości elementów tablicy. 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.*