

# Programowanie strukturalne (2023)

## - Poprawa Kolokwium 1 - Zestaw A26

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.

**Obowiązują analogiczne zasady jak na kolokwium. Maksymalna liczba punktów do zdobycia z kolokwium to 25 punktów. Nadmiarowe punkty zostaną ucięte.**

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 `calculateSeriesSum`, która przyjmuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz dodatnią liczbę wymierną  $x$ . Funkcja ma zwrócić obliczoną wartość wyrażenia:

$$1 - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1}$$

Stwórz przypadek testowy.

*Punktacja: 7 pkt.*

3. Napisz funkcję `swap_first_max`, która przyjmuje jako argumenty trzy wskaźniki na zmienne typu `int`. Funkcja powinna zamienić miejscami wartości pierwszej i największej ze zmiennych wskazywanych przez wskaźniki. Stwórz przypadek testowy w `main` tak, aby wyświetlić na konsoli wartości zmiennych przed i po zamianie. .

*Punktacja: 8 pkt.*

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

$$\begin{aligned}f_0 &= f_1 = 2 \\f_n &= f_{n-1} + 3, n\text{-parzyste}, n \geq 2 \\f_n &= f_{n-2} + 4, n\text{-nieparzyste}, n > 2\end{aligned}$$

W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy dla funkcji.

*Punktacja: 10 pkt.*

5. Napisz funkcję, która otrzymuje trzy argumenty: dodatnią liczbę całkowitą `m`, liczbę całkowitą `n` oraz `m`-elementową tablicę `tab` o elementach typu `int`. Funkcja ma zwrócić ilość elementów tablicy `tab` nie mniejszych niż `n`. W przypadku braku takich elementów zwróć 0. Stwórz dwa przypadki testowe dla funkcji.

Przykład: dla tablicy 3,4,-5,6 i `n=4` funkcja ma zwrócić 2.

*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.*