

# Programowanie strukturalne (2023)

## - Kolokwium 1 - Zestaw A04

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 sprawdzi, jaka jest największa dodatnia trzycyfrowa liczba, która jest podzielna przez podane trzy dodatnie liczby całkowite  $a$ ,  $b$  i  $c$  (podane kolejno jako argumenty funkcji). Funkcja ma zwrócić taką liczbę. Stwórz przypadek testowy dla funkcji.

*Punktacja: 7 pkt.*

3. Napisz funkcję `min_value`, która przyjmuje jako argumenty trzy wskaźniki na zmienne typu `int`. Funkcja powinna zwrócić wskaźnik na zmienną o najmniejszej wartości spośród tych wskazywanych przez wskaźniki. Stwórz przypadek testowy w main tak, aby wyświetlić na konsoli najmniejszą wartość spośród trzech zmiennych.

*Punktacja: 8 pkt.*

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

$$\begin{aligned}b_0 &= b_1 = 1 \\b_{3n} &= b_n + 1, n > 0 \\b_{3n+1} &= 2b_n, n > 0 \\b_{3n+2} &= b_n - 1, n \geq 0\end{aligned}$$

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

*Punktacja: 10 pkt.*

5. Napisz funkcję `increase_by_index`, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `int`. Funkcja ma zwiększyć wartość każdego elementu tablicy o jego indeks. 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.*