

# Programowanie strukturalne (2023)

## - Kolokwium 1 - Zestaw A10

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 liniijek czy zaburzenie struktury kodu oznacza zero punktów za polecenie.

*Punktacja: 5 pkt.*

2. Napisz program, który wczytuje ze standardowego wejścia dwie liczby całkowite  $n$  i  $m$  (zakładamy, że  $n < m$ ) i wypisuje na standardowym wyjściu liczbę elementów ciągu arytmetycznego o pierwszym wyrazie  $n$ , różnicy  $m - n$  i ostatnim wyrazie  $m$ .

*Punktacja: 7 pkt.*

3. Napisz bezargumentową funkcję `init_block_double`, która rezerwuje blok trzech zmiennych typu `double`. Funkcja ma ustawić kolejno w pamięci wartości 10.0, 20.0 i 30.0. Na koniec funkcja powinna zwrócić wskaźnik na środkową zmienną z bloku. Stwórz przypadek testowy w main tak, aby wyświetlić na konsoli wartości zmiennych przechowywanych na bloku stworzonym wewnątrz funkcji.

*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	4	...
$a_n$	2	5	7	12	19	...

Stwórz dwa przypadki testowe.

*Punktacja: 10 pkt.*

5. Napisz funkcję, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  ( $n > 3$ ) oraz  $n$ -elementową tablicę `tab` o elementach typu `double` i przesuwa o dwa w lewo wszystkie elementy tablicy (tak, żeby wartość elementu o indeksie  $n-1$  znalazła się w elemencie o indeksie  $n-3$ , wartość elementu o indeksie  $n-2$  znalazła się w elemencie o indeksie  $n-4$ , zaś wartość elementu o indeksie 0 w elemencie o indeksie  $n-2$ , a wartość elementu o indeksie 1 w elemencie o indeksie  $n-1$ ). Stwórz przypadek testowy dla funkcji.

*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.*