

# Programowanie strukturalne - Kolokwium 2 - Zestaw B65

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.
- W przypadku pominięcia danego zadania, należy dodać plik o nazwie sprecyzowanej wyżej (zawartość może być pusta).

*Za zachowanie specyfikacji dokładnie otrzymuję się dodatkowe 2 punkty. Zadania znacznie odbiegające od specyfikacji mogą nie być sprawdzane.*

1. W folderze DebugXYZ (XYZ - losowe znaki) 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. W przypadku znaków, należy zapisać sam znak w apostrofach np. `'c'` (wielkość znaków ma znaczenie).

*Punktacja: 6 pkt.*

2. Napisz funkcję, której argumentem jest napis. Funkcja ma za zadanie zwrócić liczbę znaków, które są cyframi od 0 do 4. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy.

*Punktacja: 10 pkt.*

3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiary  $n$  i  $m$ . Funkcja ma przestawić pierwszą i ostatnią kolumnę między sobą. Jeśli tablica ma mniej niż dwie kolumny, funkcja ma nic nie robić. Stwórz przypadek testowy.

Przykład.

$$\begin{bmatrix} 2 & 3 & -3 \\ 1 & 4 & 7 \\ -3 & -6 & 11 \\ -2 & 8 & 23 \end{bmatrix} \rightarrow \begin{bmatrix} -3 & 3 & 2 \\ 7 & 4 & 1 \\ 11 & -6 & -3 \\ 23 & 8 & -2 \end{bmatrix}$$

*Punktacja: 10 pkt.*

4. Stwórz typ wyliczeniowy `Vehicle` przechowujący rodzaje pojazdów. Następnie stwórz program zawierający tablicę 3 elementów typu `Vehicle`. Wypisz na konsoli zawartość tablicy używając pętli i instrukcji warunkowej.

*Punktacja: 12 pkt.*

5. Napisz funkcję, która przyjmuje jako argument listę bez głowy o elementach typu:

```
struct node {
    int num;
    struct node * next;
};
```

oraz liczbę całkowitą `d`. Funkcja ma odjąć od elementów dodatnich wartość `d`. Stwórz jeden przypadek testowy.

*Punktacja: 22 pkt.*