

# Programowanie strukturalne (2024) - Przykładowe Kolokwium 2 - Zestaw W11

Zasady kolokwium:

- Obowiązuje regulamin zajęć.
- Czas: 90 minut (ew. jak zostanie czasu do końca zajęć, to można zostać).
- Łącznie do zdobycia max 60 punktów. Próg zaliczenia: 25 pkt (bez innych punktów).
- **Kolokwium należy wykonać na komputerach zamontowanych na stałe w pracowniach.**
- Student przysyłając rozwiązania oświadcza, że rozwiązał je samodzielnie.
- W trakcie kolokwium nie można korzystać z żadnych materiałów pomocniczych w żadnej formie. Wszelkie kody powinny być napisane manualnie bez wspomagania się dodatkami automatycznie generującymi kod (np. Copilot, chat GPT itp.).
- Publikowanie poleceń i rozwiązań w internecie jest zabronione do czasu napisania kolokwium przez wszystkie grupy ćw.
- Kod musi się kompilować, aby był sprawdzany.
- Kod zakomentowany nie będzie sprawdzany.
- W trakcie kolokwium zostanie udostępniony przez prowadzącego pendrive. Zawartość pendrive będzie może zawierać pliki pomocnicze do poleceń. Udostępniony będzie w celu zgrania rozwiązań. Umieszczenie poleceń na pendrive powinno odbyć się w czasie kolokwium.
- Rozwiązania po czasie mogą nie być sprawdzane.
- O ile nie zaznaczono w poleceniu inaczej, każdą z funkcji należy wywołać co najmniej jeden raz (może być bardzo trywialnie).
- Należy przestrzegać nazw funkcji i kolejności argumentów w poleceniach.
- Warto zwracać uwagę na typ zwracany funkcji — jeśli metoda ma „coś” zwrócić, będzie to wskazane w poleceniu.
- W rozwiązaniach nie należy wykonywać nadmiarowych czynności, niewskazanych w poleceniu w ramach samodzielnie zdefiniowanej funkcji. Dodatkowe czynności mogą odbywać się w main. O ile w poleceniu nie zaznaczono inaczej, w rozwiązaniu można stworzyć funkcje pomocnicze.
- Po kartkach z poleceniami można pisać i traktować jako brudnopis.
- Format rozwiązania:
  - każde rozwiązania w osobnym pliku z rozszerzeniem.
  - nazwa plików: zad1.c, zad2.c, zad3.c, zad4.c, zad5.c.
  - 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

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: 6 pkt.*

2. Napisz funkcję, której argumentem jest napis oraz znak. Funkcja ma zwrócić ile razy znak występuje w napisie. Stwórz przypadek testowy.

Przykład: dla napisu "agrafka" i znaku 'a' funkcja powinna zwrócić 3.

*Punktacja: 10 pkt.*

3. Napisz funkcję, której argumentem jest dwuwymiarowa kwadratowa tablica tablic (zawierająca zmienne typu int) oraz jej wymiar  $n$ . Funkcja ma zwrócić indeks wiersza, w którym znajduje się największy element - w przypadku kilku takich wierszy, zwróć indeks ostatniego z nich. Stwórz przypadek testowy.

Przykład: Dla tablicy

$$\begin{bmatrix} 3 & -4 & 5 \\ 1 & 22 & -33 \\ -3 & 5 & 22 \end{bmatrix}$$

funkcja powinna zwrócić dwa.

Tablica tablic powinna być zadeklarowana jako podwójny wskaźnik.

*Punktacja: 10 pkt.*

4. Stwórz strukturę `Movie` o dwóch polach: `title` (napis) i `duration` (int). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Movie` oraz rozmiar tablicy. Funkcja ma zwrócić film (jako strukturę) o najkrótszym czasie trwania. W przypadku kilku filmów w tablicy o najkrótszym czasie trwania, to zwróć pierwszą z możliwych. Stwórz przypadek testowy.

Przykład: dla tablicy struktur:

Indeks	title	duration
0	Ida	5
1	Bogowie	12
2	Oblawa	7
3	Sami swoi	5

funkcja powinna zwrócić strukturę o tytule Ida i długości 5.

*Punktacja: 12 pkt.*

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

```
struct element {  
    int x;  
    struct element * next;  
};
```

Funkcja ma wyświetlić te elementy listy, które są większe od pięciu. Stwórz przypadek testowy.

Wskazówka: Lista bez głowy ma tyle struktur ile ma elementów. Adres listy to adres pierwszego elementu. Pusta lista to `NULL`.

Przykład: dla listy o elementach 3,-9,11,-3,9,5,12 powinno zostać wyświetlone 11,9,12.

*Punktacja: 22 pkt.*

