

# Programowanie strukturalne (2024) - Przykładowe Kolokwium 2 - Zestaw W14

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 rozwiązanie 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 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 są dwa napisy. Funkcja powinna zwrócić informację ile znaków będących cyframi jest w krótszym napisie. Jeśli napisy są równej długości, to funkcja powinna zwrócić liczbę znaków cyfr z pierwszego napisu. Stwórz przypadek testowy

Przykład: dla napisów "abc2" oraz "xyz345" funkcja powinna zwrócić 1.

*Punktacja: 10 pkt.*

3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiary  $n$  i  $m$ . Funkcja ma odwrócić kolejność elementów w wierszach o parzystych indeksach. Stwórz przypadek testowy.

Przykład.

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

Wskazówka: Tablica tablic powinna być zadeklarowana jako podwójny wskaźnik.

*Punktacja: 10 pkt.*

4. Stwórz strukturę `Szkola` o dwóch polach `typ (int)` oraz `numer (int)`. Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Szkola` oraz rozmiar tablicy. Funkcja ma zwrócić największy spośród numerów szkół. Stwórz przypadek testowy.

Przykład: dla tablicy struktur:

Indeks	typ	numer
0	3	44
1	2	23
2	5	38
3	4	42

funkcja powinna zwrócić 44.

*Punktacja: 12 pkt.*

5. Napisz funkcję `printNegative`, która przyjmuje jako argument listę z głową o elementach typu:

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

Funkcja ma wyświetlić te elementy listy, które są ujemne. Stwórz przypadek testowy.

Wskazówka: Lista z głową ma pierwszy element, którego wartość nie jest zainicjowana. Jeśli lista jest pusta (ma samą głowę), to funkcja nie powinna nic robić. Dla listy z elementami 3,-8,12,-4 funkcja ma wyświetlić -8 i -4.

*Punktacja: 22 pkt.*

