

# Programowanie strukturalne (2024) - Przykładowe Kolokwium 1 - Zestaw S11

Zasady kolokwium:

- Obowiązuje regulamin zajęć.
- Czas: 90 minut (ew. jak zostanie czasu do końca zajęć, to można zostać).
- Łącznie do zdobycia max 40 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.
- 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: 5 pkt.*

2. Napisz funkcję, która obliczy przybliżoną wartość liczby Eulera  $e$  na podstawie sumy  $n$  pierwszych wyrazów szeregu:  $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$ . Funkcja przyjmuje jako argument dodatnią liczbę naturalną  $n$  i zwraca wartość przybliżonej liczby Eulera. Stwórz przypadek testowy.

*Punktacja: 7 pkt.*

3. Napisz funkcję `linear_function_value`, która ma trzy argumenty. Pierwszym argumentem jest wskaźnik `a` na stałą wartość typu `float`, drugim argumentem jest stały wskaźnik `b` na zmienną typu `float`, a trzecim argumentem jest wskaźnik `x` na stałą wartość typu `float`. Funkcja `linear_function_value` ma zwrócić wartość funkcji liniowej o współczynnikach wskaźnianych przez wskaźniki `a` i `b` dla argumentu `x` ( $y=ax+b$ ). Stwórz przypadek testowy dla funkcji.

*Punktacja: 8 pkt.*

4. Napisz rekurencyjną funkcję `sum_of_digits`, która zwraca sumę cyfr otrzymanej w argumencie nieujemnej liczby całkowitej  $n$ . Stwórz dwa przypadki testowe.

*Punktacja: 10 pkt.*

5. Napisz funkcję `count_positive_elements`, która otrzymuje dwa argumenty: dodatnią liczbę całkowitą  $n$  oraz  $n$ -elementową tablicę `tab` o elementach typu `int`. Funkcja ma zwrócić liczbę dodatnich elementów w tablicy. Stwórz przypadek testowy.

Przykład. Dla tablicy zawierającej elementy 3,-4,5 powinno być zwrócone 2.

*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.*

