

# Zadania w C++

## Zadanie 1: Silnia.

Silnia: Napisz program, zawierający definicję funkcji silnia. Silnia liczby całkowitej (n) oznacza iloczyn wszystkich liczb całkowitych od 1 do (n). Możesz użyć pętli for, while lub rekurencji.

**Dane wejściowe:** n

**Dane wyjściowe:** silnia(n)

## Zadanie 2: Układ równań liniowych.

Napisz program, który rozwiązuje układ dwóch równań liniowych w postaci:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

**Dane wejściowe:**  $a_1, b_1, c_1, a_2, b_2, c_2$

**Dane wyjściowe:** x, y

## Zadanie 3: Średnia arytmetyczna.

Napisz program, który oblicza średnią arytmetyczną liczb podanych z klawiatury. Liczby podawane są jedna obok drugiej (lub jedna pod drugą) i trafiają do tablicy o maksymalnym rozmiarze 100. Program zakończy pobieranie danych po wprowadzeniu wartości -1 lub po osiągnięciu limitu bufora. Następnie dane są wypisywane na ekran, jedna obok drugiej, a w kolejnej linii wyświetlana jest średnia arytmetyczna. Średnia arytmetyczna to suma liczb podzielona przez ich ilość.

**Dane wejściowe:** kolejno wprowadzane wartości

**Dane wyjściowe:** wprowadzone dane i średnia arytmetyczna

## Zadanie 4: Sortowanie liczb.

Sortowanie liczb: Napisz program, który sortuje trzy liczby rosnąco. Do tego celu zdefiniuj funkcję zamieniającą wartości miejscami.

**Dane wejściowe:** a, b, c

**Dane wyjściowe:** posortowane liczby (a, b, c).

## Zadanie 5: Liczby pierwsze.

Napisz program, który wypisuje liczby pierwsze od 2 do n. Liczba pierwsza to taka, która ma dokładnie dwa dzielniki: 1 i siebie samą.

**Dane wejściowe:** n

**Dane wyjściowe:** liczby pierwsze z zakresu  $<2, n>$

## Zadanie 6: Ciąg Fibonacciego.

Napisz program, który wypisuje pierwsze (n) liczb z ciągu Fibonacciego. Ciąg ten zaczyna się od 0 i 1, a każdy kolejny element jest sumą dwóch poprzednich.

**Dane wejściowe:** n

**Dane wyjściowe:** ciąg Fibonacciego.

## Zadanie 7: Obliczanie pola trójkąta.

Napisz program, który oblicza pole trójkąta na podstawie długości jego boków. Możesz użyć wzoru Herona.

**Dane wejściowe:**  $a, b, c$

**Dane wyjściowe:**  $PoleT(a, b, c)$

## Zadanie 8: Szukanie maksimum i minimum.

Napisz program, który znajduje maksymalną i minimalną wartość w tablicy liczb całkowitych o rozmiarze  $n$  wypełnionej wartościami losowymi. Użyj dynamicznej alokacji tablicy.

**Dane wejściowe:**  $a, b, n$

**Dane wyjściowe:** *Lista wygenerowanych liczb, min, max*

## Zadanie 9: Klasa ArrayOperations

Napisz klasę ArrayOperations, która będzie miała metody do znajdowania sumy, średniej i maksymalnego i minimalnego elementu w dynamicznej tablicy liczb całkowitych. Klasa powinna zawierać prywatny wskaźnik do przechowywanych elementów oraz publiczny konstruktor tworzący dynamicznie alokowaną tablicę elementów, destruktor i metodę do dodawania kolejnych elementów tablicy.

## Zadanie 10: Odwracanie tablicy

Zdefiniuj klasę, która będzie zawierała dynamiczną tablicę i metodę do odwracania jej elementów. Klasa powinna zawierać prywatny wskaźnik do przechowywanych elementów oraz publiczny konstruktor tworzący dynamicznie alokowaną tablicę elementów, destruktor i metodę do dodawania kolejnych elementów tablicy.

## Zadanie 11: Liczenie wystąpień elementów

Zdefiniuj klasę, która będzie zawierała dynamiczną tablicę i metodę do zliczania ilości wystąpień jej elementów. Klasa powinna zawierać prywatny wskaźnik do przechowywanych elementów oraz publiczny konstruktor tworzący dynamicznie alokowaną tablicę elementów, destruktor i metodę do dodawania kolejnych elementów tablicy.

## Zadanie 12: Suma dwóch tablic

Napisz funkcję, która dodaje elementy dwóch tablic liczb całkowitych o tej samej długości i zwraca nową tablicę z wynikami.

## Zadanie 13: Znajdowanie indeksu elementu

Napisz funkcję, która znajduje indeks pierwszego wystąpienia danego elementu w tablicy liczb całkowitych. Tablica o rozmiarze  $n$  powinna być wypełniona wartościami losowymi.

**Dane wejściowe:**  $n$

**Dane wyjściowe:**  $x, index$

## Zadanie 14: Klasa StringArray

Napisz klasę StringArray, która będzie przechowywała tablicę łańcuchów znaków i będzie miała metodę do łączenia wszystkich łańcuchów w jeden ciąg.

## Zadanie 15: Implementacja klasy DynamicArray

Napisz klasę DynamicArray, która będzie implementowała dynamiczną tablicę. Klasa powinna mieć metody do dodawania, usuwania i przeglądania elementów. Napisz program, który przetestuje działanie tej klasy.

## Zadanie 16: Algorytm wyszukiwania binarnego

Napisz funkcję implementującą algorytm wyszukiwania binarnego w posortowanej tablicy. Tablica powinna zawierać  $n$  losowych wartości ułożonych od najmniejszej do największej wartości.

**Dane wejściowe:**  $n$ , *szukany\_element*

**Dane wyjściowe:** *index lub -1 jeżeli element nie zostanie znaleziony*

## Zadanie 17: Permutacje tablicy

Napisz funkcję, która generuje wszystkie permutacje tablicy liczb całkowitych o rozmiarze 5, wypełnioną wartościami losowymi z zakresu  $<a, b)$ . Przetestuj działanie na małych tablicach.

**Dane wejściowe:**  $a, b$ .

**Dane wyjściowe:** *wszystkie permutacje tablicy.*

## Zadanie 18: Anagramy

Napisz funkcję, która sprawdza, czy dwie tablice znaków (łańcuchy) są anagramami.

**Dane wejściowe:** *tekst1, tekst2*

**Dane wyjściowe:** *tak/nie*