

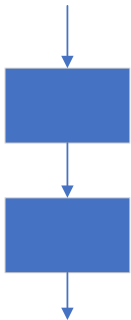
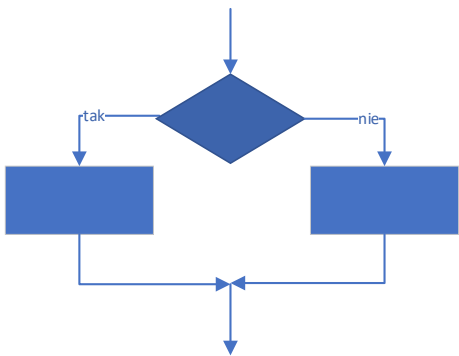
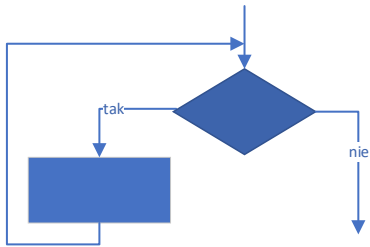
ALGORYTMY I STRUKTURY DANYCH

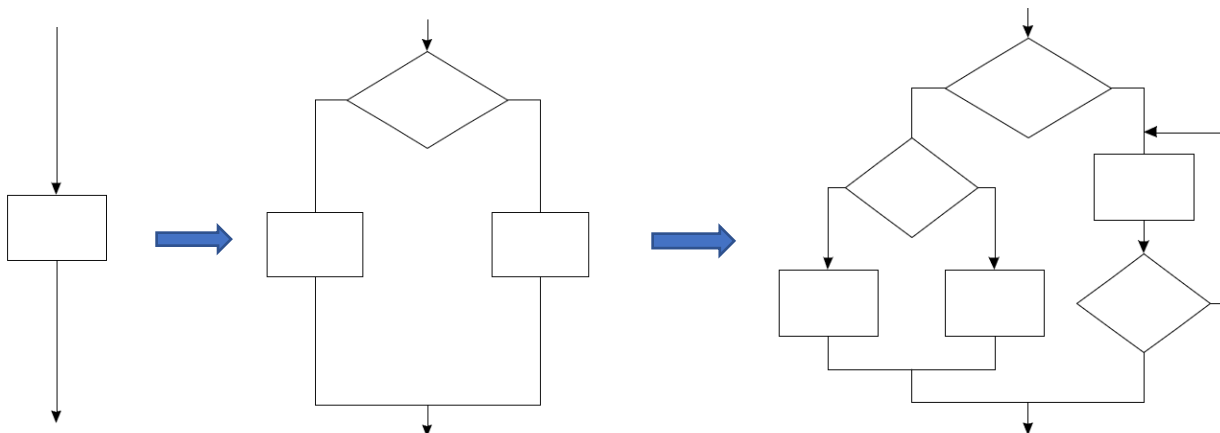
LABORATORIUM 1:

ALGORYTMY WYSZUKIWANIA I SORTOWANIA

Programowanie strukturalne polega na budowaniu algorytmów (i programów) z użyciem tylko trzech konstrukcji:

- sekwencyjnej
- warunkowej
- iteracyjnej (ograniczonej i warunkowej)

		
<pre>instrukcja1; instrukcja2;</pre>	<pre>if (warunek) instrukcja1; else instrukcja2;</pre>	<pre>while (warunek) { instrukcja; }</pre>
<pre>x = 5; x = x + 10;</pre>	<pre>if (a % 2 == 0) flaga = true; else flaga = false;</pre>	<pre>while (i < 10) { i++; }</pre>



Aplikacja do rysowania schematów blokowych: <https://app.diagrams.net/>

Zadanie 1. Narysuj schemat blokowy algorytmu, który sprawdza czy podana przez użytkownika wartość występuje w tablicy jednowymiarowej.

(Pliki do wykorzystania: [zadanie1.cpp](#))

Zadanie 2. Zaprojektuj algorytm wyszukiwania w tablicy jednowymiarowej minimalnej wartości.

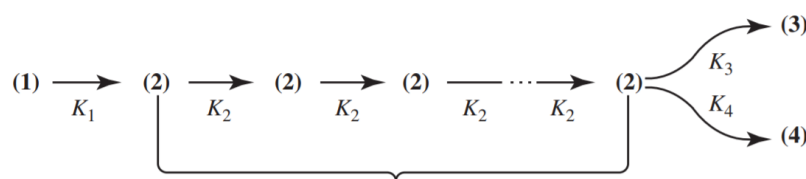
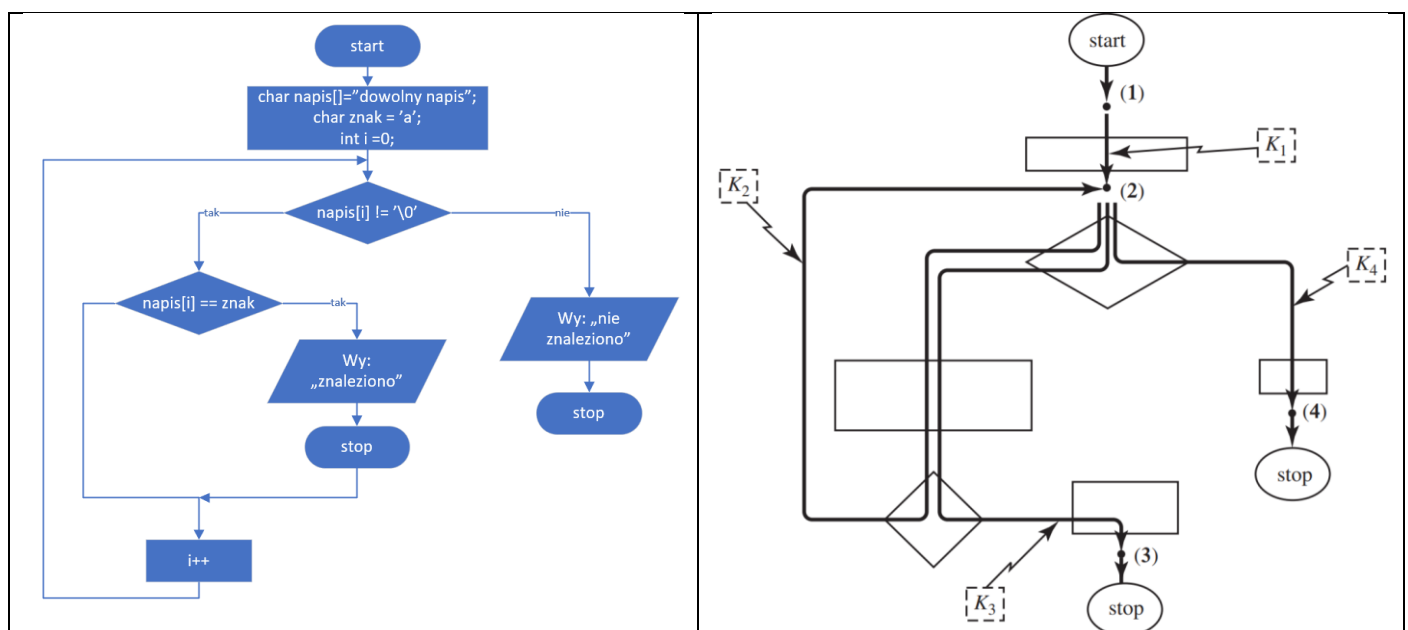
Zadanie 2*. Zaprojektuj algorytm wyszukiwania w tablicy dwuwymiarowej minimalnej wartości w każdym wierszu. Po znalezieniu minimalnej wartości wstaw ją na początek danego wiersza (poprzez zamianę miejsc).

Zadanie 3. Wykonaj analizę działania algorytmu sortowania bąbelkowego na przykładowych danych. Dodatkowo*: Zaimplementuj algorytm sortowania bąbelkowego, w którym po stwierdzeniu braku przestawień elementów tablicy, algorytm kończy działanie.

(Pliki do wykorzystania: [Lab1_zadania.xlsx](#), [arkusz zadanie_3](#); [zadanie3_bubbleSort.cpp](#))

Złożoność obliczeniowa określa liczbę instrukcji wykonywaną przez program dla zadanego zbioru danych wejściowych.

Złożoność pamięciowa jest to liczba komórek pamięci potrzebnych do rozwiązania dowolnego zadania rozmiaru n .



D. Harel „Rzecz o istocie informatyki”

Na rysunku stałe K_1 do K_4 są związane z czterema możliwymi drogami. Zakłada się, iż każde wykonanie lokalnego odcinka obejmuje nie więcej niż K_i instrukcji.

Zadanie 4. Wykonaj analizę złożoności obliczeniowej algorytmu sortowania bąbelkowego.

(Pliki do wykorzystania: [bubble_sort_complexity.pdf](#))

Zadanie 5. Wykonaj analizę działania algorytmu sortowania przez wybór na przykładowych danych.

(Pliki do wykorzystania: [Lab1_zadania.xlsx](#), [arkusz zadanie_5](#))

Zadanie 6. Wykonaj analizę działania algorytmu sortowania przez wstawianie na przykładowych danych.

(Pliki do wykorzystania: [Lab1_zadania.xlsx](#), [arkusz zadanie_6](#))

Zadanie 7. Wykonaj analizę złożoności obliczeniowej algorytmu sortowania przez wstawianie.
(Pliki do wykorzystania: insert_sort_complexity.pdf)

Zadanie 8*. Napisz program sortowania **liczb** naturalnych „leksykograficznie” względem rozwinięć dziesiętnych.

Przykład: Wejście: 1, 2, 3, 11, 21, 111, 231

 Wyjście: 1, 11, 111, 2, 21, 231, 3

Zadanie 9*. Napisz program sortowania wierszy macierzy „leksykograficznie”. Zastosuj algorytm sortowania przez wybór (selectionSort) lub przez wstawianie (insertSort).