ALGORYTMY I STRUKTURY DANYCH – WPROWADZENIE

- **Algorytm**
  - zbiór czynności (kroków) prowadzących do wykonania pewnego zadania (rozwiązania problemu).

- **Podstawowe cechy dobrego algorytmu**
  - skończoność (własność stopu), dobre zdefiniowanie (każdego kroku i zbioru danych), poprawność, szybkość, prostoto ta, …

- **Reprezentacje algorytmów**
  - Lista kroków

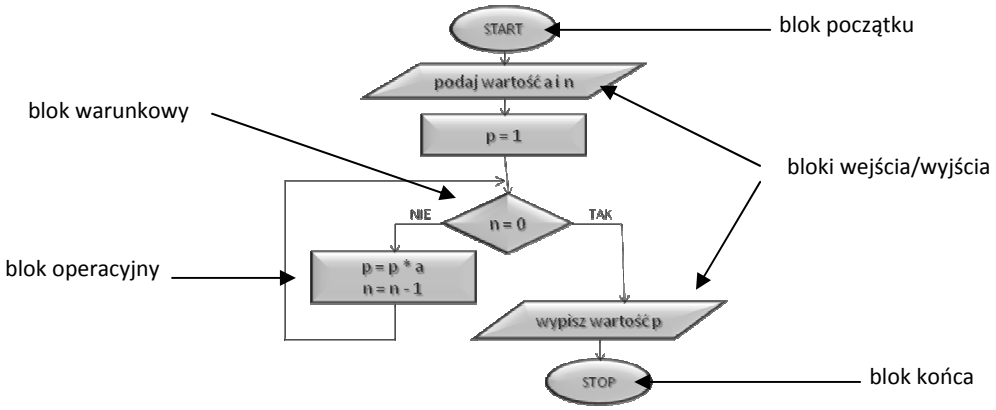
    Przykłady:

    1. *wczytaj wartość danej x*
    2. *f(x) = 2*x + 1*
    3. *wypisz wartość f(x)*
    4. *zakończ algorytm*

    1. *wczytaj wartość danych a i n*
    2. *jeśli a < 0 lub n < 0 to wypisz „błąd danych” i przejdź do kroku 8*
    3. *p = 1*
    4. *jeśli n = 0 to przejdź do kroku 7*
    5. *p = p * a*
    6. *n = n -1 i przejdź do kroku 4*
    7. *wypisz wartość p*
    8. *zakończ algorytm*

  - Schemat blokowy (sieć działań)

    Przykład:



    - Inne bloki
      - łączniki stronnicowe i między stronnicowe
      - informacyjne (komentarz)
      - podprogramu

  - Pseudokod i język programowania
    - pseudokod – notacja oparta na języku programowania (Pascal, C), niezależny od technik programowania, wykorzystujący język naturalny i notację matematyczną w celu uproszczenia zapisu, ale z zachowaniem cech języka programowania

    Przykłady:

1. *wczytaj a i n*
2. *p ← 1*
3. *while (n > 0)*
4. *p ← p * a*
5. *n ← n - 1*
6. *wypisz p*

```
2.  #include <stdio.h>
3.
4.  main()
5.  {
6.    int a, n, p;
7.    scanf(”%d%d”, &a, &n);
8.    p = 1;
9.    while (n > 0)
10.   {
11.     p = p * a;
12.     n --;
13.   }
14.   printf(”%d%d”, &a)
```