



Algorytmy i struktury danych

Łukasz Bednarski



Etapy rozwiązywania problemu

W jaki sposób rozwiązać określony problem?

1. Sformułowanie zadania - problemu.
2. Określenie danych wejściowych.
3. Określenie celu, czyli wyniku.
4. Poszukiwanie metody rozwiązania, czyli algorytmu.
5. Przedstawienie algorytmu w postaci: opisu słownego, listy kroków, schematu blokowego, jednego z języków programowania
6. Analiza poprawności rozwiązania.
7. Testowanie rozwiązania dla różnych danych - ocena efektywności przyjętej metody.



Algorytm to skończony, uporządkowany zbiór jasno zdefiniowanych czynności, koniecznych do wykonania pewnego zadania, w ograniczonej liczbie kroków.

Algorytm jest to przepis na rozwiązanie określonego problemu.

Przykład: Algorytm gotowania jajka na miękko.

- Krok 1. Włóż jajko do gotującej się wody.
- Krok 2. Zanotuj czas początkowy t_0 .
- Krok 3. Odczytaj czas aktualny t .
- Krok 4. Oblicz $D t = t - t_0$.
- Krok 5. Jeśli $D t < 3 \text{ min.}$, to przejdź do kroku 3.
- Krok 6. Wyjmij jajko z gotującej się wody. Zakończ algorytm.

Cechy algorytmu

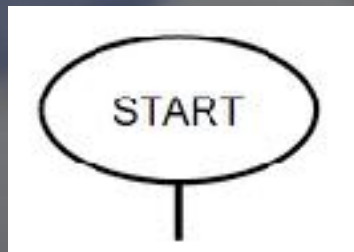


- poprawność
 - algorytm daje oczekiwane wyniki
- jednoznaczność
 - zawsze daje te same wyniki przy takich samych danych wejściowych
- skończoność
 - wykonuje się w skończonej liczbie kroków
- sprawność
 - czasowa - szybkość działania i pamięciowa

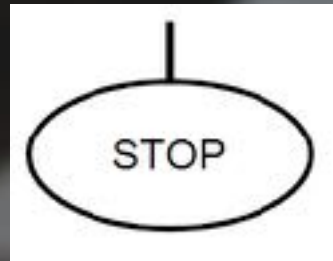


1. **Słowny opis** – Rozpoczyna się dyskusją w jaki sposób można rozwiązać dane zadanie. Służy wyrobieniu intuicji i ukierunkowaniu rozwiązań we właściwe sposoby i techniki przydatne w rozwiązaniu.
2. **W postaci listy kroków** - jest to dokładny sposób opisywania obliczeń i ich kolejności. Kroki zawierają opis operacji, które mają być wykonane przez algorytm. Występują tu polecenia związane ze zmianą kolejności wykonania kroków lub polecenia zakończenia algorytmu.
3. **Schemat blokowy** – najpopularniejszy; graficzny zapis który składa się z bloków oraz połączeń między nimi. Są tu zapisane operacje, które mają być wykonane, a połączenia wyznaczają kolejność wykonania.

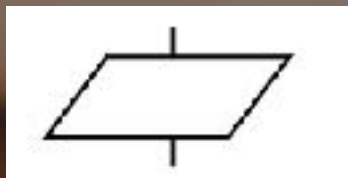
Schemat blokowy



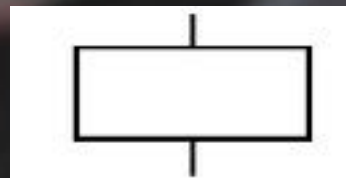
Blok początkowy (start) rozpoczyna algorytm. Ma tylko jedno połączenie wychodzące.



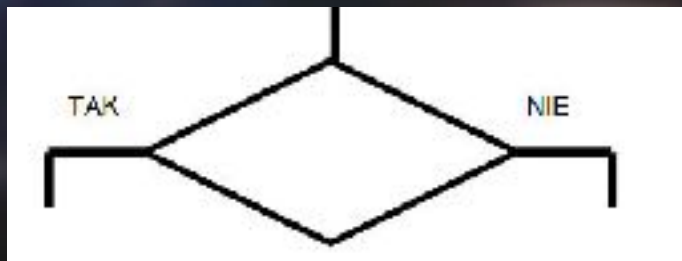
Blok końcowy (koniec) kończy działanie algorytmu. Nie wychodzą z niego żadne połączenia.



Blok WE/WY (wejścia/wyjścia) jest równoległobokiem, w którym umieszcza się dane lub wyniki.



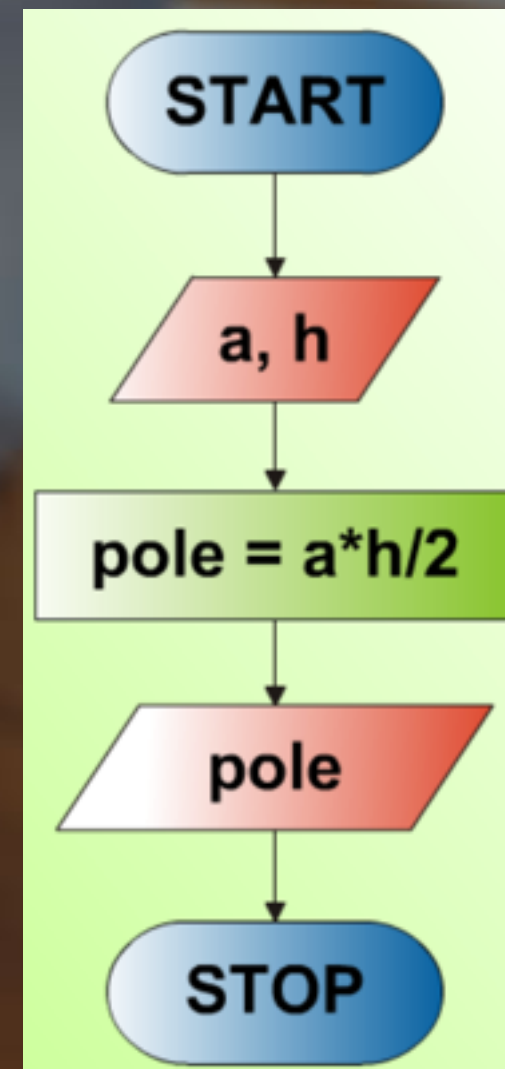
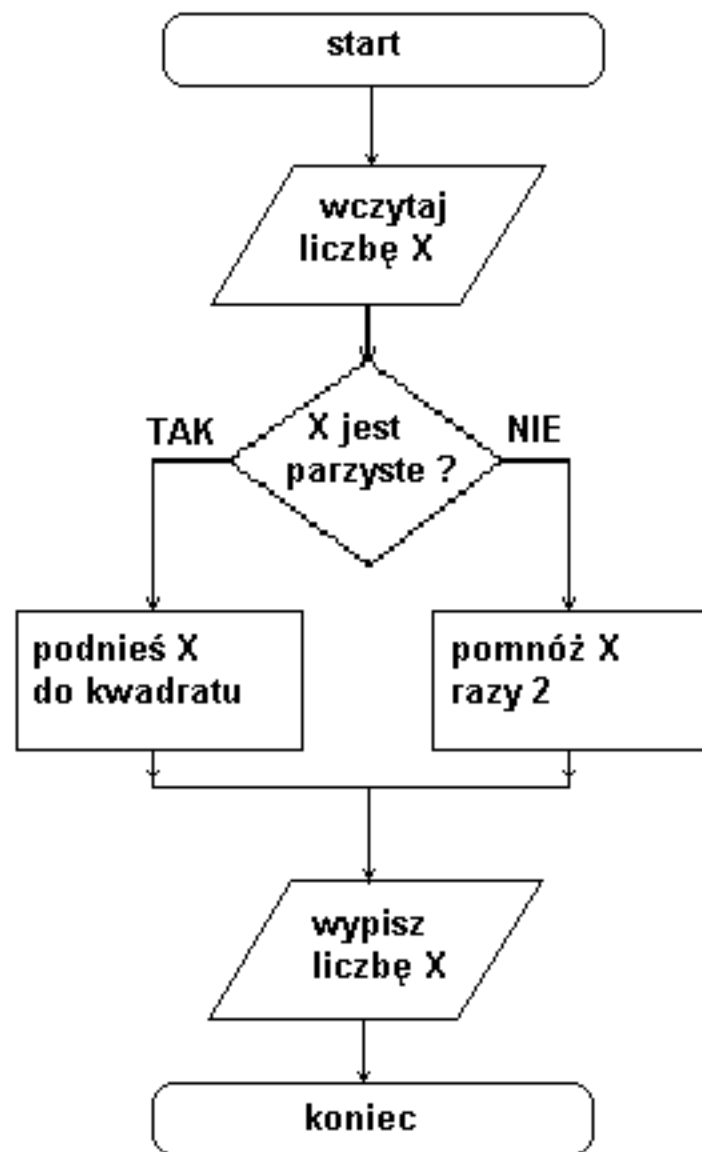
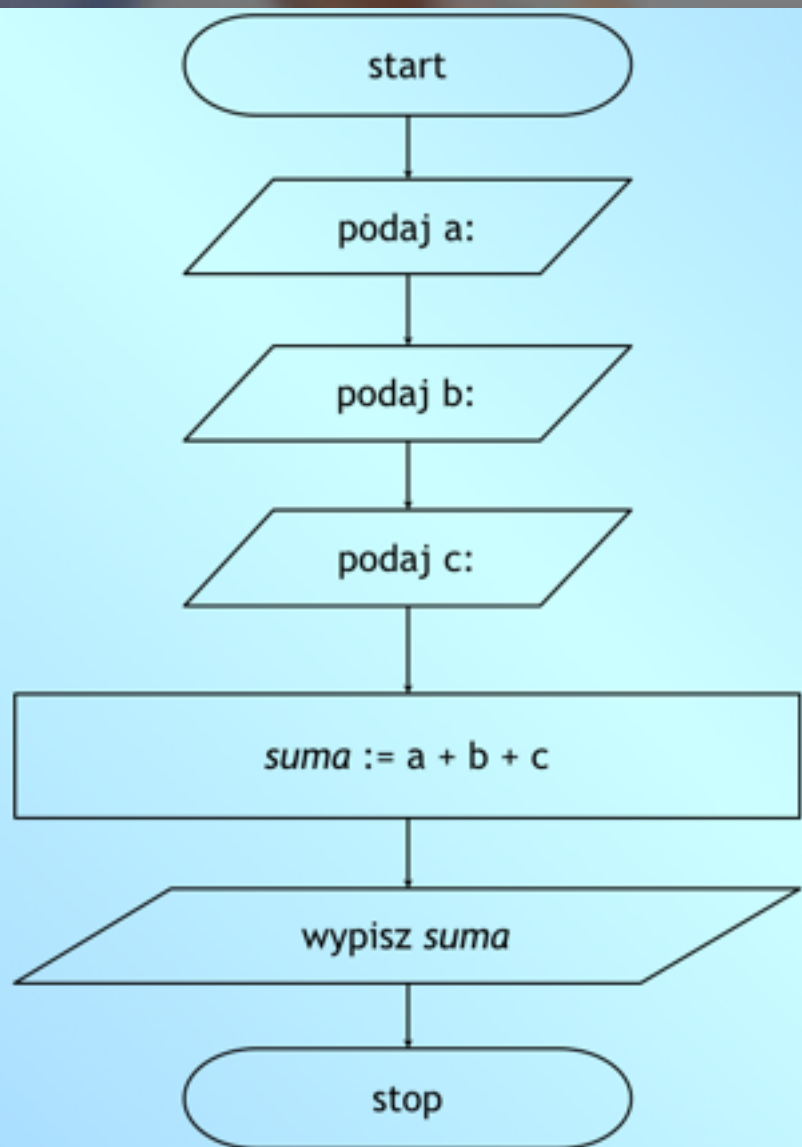
Blok operacyjny (wykonywanie operacji, czyli konkretnych działań). Można w nim wpisać więcej niż jedno wyrażenie.



Blok warunkowy (decyzyjny) jest rombem, w którym umieszcza się warunek decydujący o dalszej kolejności wykonywania operacji.

- „Tak”, jeśli warunek jest spełniony
- „Nie”, jeśli warunek nie jest spełniony

Schemat blokowy





1. **Algorytmy sekwencyjne (liniowe)** to algorytmy, w których wszystkie kroki wykonuje się w kolejności takiej, w jakiej zostały zapisane (np. obliczanie wielomianów); klocki w schemacie blokowym takiego algorytmu są ustawione jeden pod drugim.
2. **Algorytmy z rozgałęzieniami** to coś podobnego do algorytmów liniowych, z tym, że są różne drogi (rozgałęzienia) do uzyskania wyniku - inaczej mówiąc, algorytm wykonywany jest w zależności od pewnych warunków.
3. **Algorytmy rekurencyjne** to algorytmy, w których występuje funkcja/procedura wywołująca samą siebie, wywoływanie takie kończy się spełnieniem jakiegoś warunku. Charakterystyczną cechą funkcji (procedury) rekurencyjnej jest to, że wywołuje ona samą siebie (np. silnia).
4. **Algorytmy iteracyjne** charakteryzują się zapętleniem pewnych czynności, które są powtarzane aż do uzyskania jakiegoś celu; wówczas zostaje spełniony warunek kończący działanie pętli.

Zadanie



1. Wprowadź liczbę dodatnią całkowitą i sprawdź czy podana liczba jest parzysta.
2. Rozbuduj algorytm aby przyjąć 3 liczby z klawiatury i zsumować tylko liczby parzyste.
3. Rozbuduj algorytm aby przyjmować 5 liczb z klawiatury. Parzyste powinny być sumowane, natomiast nieparzyste odejmowane. Wyświetl listę wprowadzonych liczb oraz wynik.
4. Utwórz algorytm który w zdefiniowanej 10 elementowej tablicy policzy i wyświetli średnią.
5. Utwórz algorytm który w 10 elementowej tablicy znajdzie i wyświetli liczbę największą oraz najmniejszą. Następnie spróbuj zaimplementować też aby zwracany był indeks podanej liczby.
6. Napisz program który będzie przyjmował od użytkownika liczby do momentu podania 0. Jeśli użytkownik wpisze 0, wprowadzanie danych powinno się zakończyć i powinna zostać wyświetlona liczba wszystkich elementów, liczb parzystych oraz nieparzystych.
7. Rozbuduj algorytm w taki sposób aby po zakończeniu przyjmowania danych, wyświetlone zostały użytkownikowi wszystkie liczby a w kolejnych liniach parzyste i nieparzyste.
8. Utwórz algorytm który pobierze od użytkownika słowo, wyświetli je w połowie oraz długość.
9. Utwórz algorytm który sprawdzi czy słowo jest palindromem i zwróci TAK lub NIE.
10. Napisz program który w podanych ciągu słów (jakieś zdanie) znajdzie i wyświetli najdłuższe i najkrótsze słowo oraz słowo najczęściej i najrzadziej występujące (jeśli jakieś słowa wystąpiły tyle samo razy i są najczęściej lub najrzadziej pojawiającymi się, wyświetl każde z nich).

Zadanie



1. Utwórz algorytm liczący silnię.
2. Utwórz algorytm rekurencyjny liczący silnię.
3. Utwórz algorytm obliczający ciąg Fibonacciego.
4. Napisz algorytm obliczający NWD - największy wspólny dzielnik.
5. Utwórz algorytm iteracyjny sprawdzający czy podana liczba jest liczbą pierwszą.
6. Utwórz algorytm sprawdzający czy podany PESEL posiada poprawny format.