Python

Algorytmy i struktury danych

Algorytmy – ćwiczenia

Algorytmy

- **Algorytm – przepis na rozwiązanie problemu obliczeniowego.**
- Różnica między algorytmem a wzorcem projektowym polega na tym, że ten drugi rozwiązuje problem architektury oprogramowania, podczas gdy algorytm skupia się na obliczeniach.
- **Algorytm – ciąg operacji przeprowadzony na różnych strukturach danych zmierzający do rozwiązania problemu programistycznego.**

Algorytmy – sposób postępowania

1. **Sformułowanie zadania** – ustalamy jaki problem ma rozwiązywać algorytm.
2. **Dane wejściowe** – czego oczekujemy, co powinniśmy dostać na wejściu, jaki typ, wartości, ograniczenia.
3. **Określenie wyniku** – co powinien zwrócić algorytm (napis? listę? funkcję?)
4. **Ustalenie metody wykonania zadania** – tutaj może się pojawić kilka sposobów na rozwiązanie jednego problemu (to my ustalamy, który jest dla nas najodpowiedniejszy).
5. **Zapisanie algorytmu za pomocą wybranej metody.**
6. **Analiza i testowanie zaproponowanego algorytmu i jego działania.**
7. **Ocena skuteczności algorytmu** – efektywność, prostota, łatwość implementacji; niech świat się dowie i nas osądzi ☺

Pierwszy powtarzający się znak

Zaproponuj algorytm i napisz funkcję, która przyjmować będzie jeden argument typu string.
Zadaniem funkcji będzie wskazanie litery, która jako pierwsza zostanie powtórzona w stringu (dla uproszczenia przyjmujemy, że napis będzie posiadał tylko małe litery).

Przykład: ’’akapit’’ -> ’a’

Najdłuższy ciąg znaków

Zaproponuj algorytm i napisz funkcję, która przyjmować będzie jeden argument typu string, np. ’’AABCCDDDEF’’. Zadaniem funkcji będzie wskazanie litery, która posiada najdłuższy ciąg pojawiający się po sobie. W przykładowym napisie, jest to znak ’D’, ponieważ występuje 3 razy po sobie i jest to najdłuższy taki ciąg w całym stringu.

Liczba pi

Oblicz wartość liczby π.

Algorytmy

Różne sposoby prezentacji algorytmów:

- Słowny
- Schemat blokowy
- Pseudokod
- Kod

1. Wlej wodę do garnka
2. Włóż do garnka x jajek
3. Zagotuj wodę
4. Gotuj jajka 8 minut
5. Wylej wodę z garnka
6. Schłódź jajka zimną wodą


```
In [20]: def add(a, b):
    ...:     return a + b
    ...:
```

jeżeli numer karty kredytowej jest ważny **to**
    wykonanie transakcji w oparciu o numer karty i zamówienie
w przeciwnym razie
    wyświetlenie wiadomości o niepowodzeniu
koniec warunku

Sposoby opisywania algorytmów

Zaproponuj słowny opis algorytmu dodawania do siebie trzech liczb.

Czy było to trudne? Długie? Żmudne?

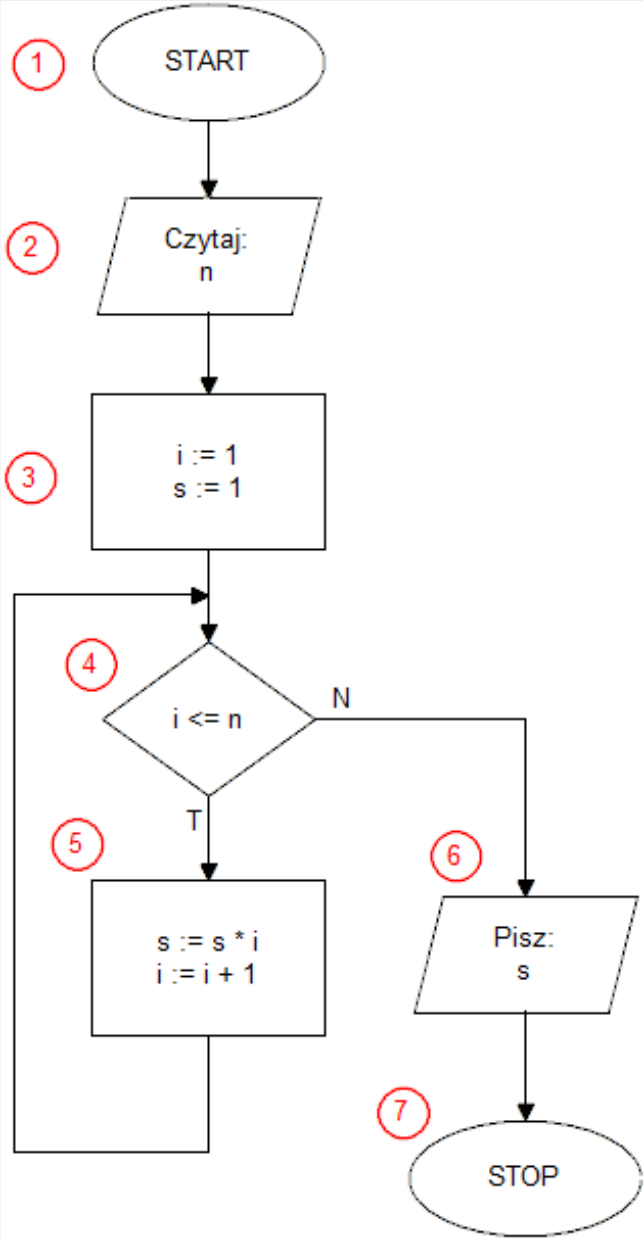
A co w przypadku słownego opisania algorytmu obliczania liczby pi? NWD? Zamiany liczby dziesiętnej na binarną?

Schematy blokowe

Schematy blokowe

Schemat blokowy to graficzny zapis algorytmu na porzeby rozwiązania konkretnego zadania. Na jego podstawie ustalamy/odczytujemy kolejność wykonywanych instrukcji i przepływ algorytmu.

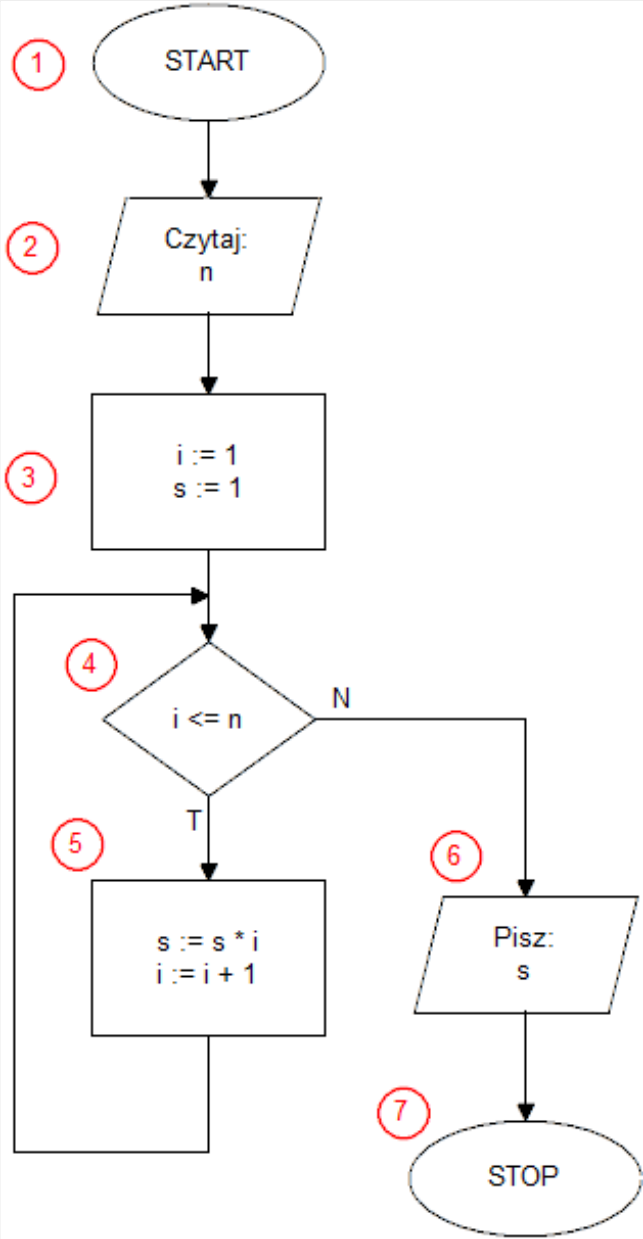
- proste w użyciu – reprezentują kolejne czynności w algorytmie,
- prosta budowa – mała ilość elementów (bloków),
- można je tworzyć z wykorzystaniem zapisu składniowego dowolnie wybranego języka (różne operatory, np. := vs =),
- przydatne w przypadku tworzenia zaawansowanych algorytmów,
- na ich podstawie dużo prościej o napisanie kodu danego algorytmu



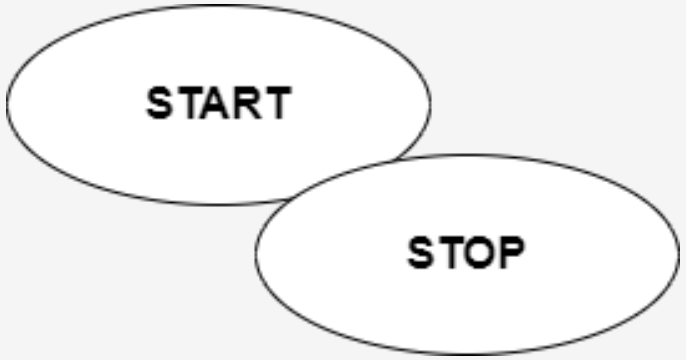
Schematy blokowe

Wyróżniamy kilka rodzajów schematów (związane jest to bezpośrednio z użytymi blokami/stopniem zaawansowania schematu):

- **Proste (sekwencyjne)** - nie używa się w nich bloków warunkowych. W takiej sieci działań kolejność realizacji poszczególnych operacji jest ściśle określona i żadna z nich nie może być pominięta ani powtórzona.
- **Z rozwidleniem** - zawiera w sobie wybór jednej z kilku możliwych dróg realizacji danego zadania. Istnieje w nim przynajmniej jeden blok warunkowy.
- **Z pętlą** - często w trakcie realizacji danego zadania konieczne jest powtórzenie niektórych operacji różniących się jedynie zestawem danych. Pętla obejmuje tę część bloków, która ma być powtarzana.
- **Złożone** - będące kombinacją powyższych sieci.



Bloki graniczne



Są to bloki w kształcie owalu, od których rozpoczyna się i na któych kończy reprezentację blokową algorytmu. Z bloku START (zazwyczaj tak nazywanego) wychodzi tylko i wyłącznie jedna strzałka w dół, nie wchodzi za to żadna. Do bloku STOP (KONIEC itd.) wchodzi jedna strzałka, natomiast nie wychodzi już żadna – w momencie kiedy algorytm dotrze do tego bloku, kończy się jego wykonywanie.

Funkcja: oznaczenie początku i końca algorytmu

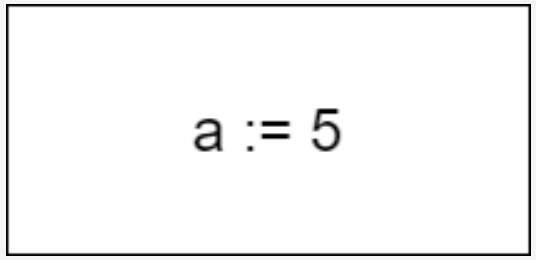
Schematy blokowe - symbole

Strzałka/łącznik

Strzałka (łącznik) wskazuje jednoznacznie kierunek przepływu instrukcji – nigdy nie powinna się rozwidlać, algorytm powinien jasno przechodzić z jednego bloku do drugiego. Należy jednak pamiętać, że możliwy jest powrót za pomocą strzałki z bloku poniżej do bloku wyżej (mamy wtedy do czynienia z zapętleniem instrukcji).

Funkcja: wskazuje kolejność wykonywania się instrukcji w algorytmie

Skrzynka operacyjna



Skrzynka operacyjna jest reprezentowana przez prostokąt. W jej wnętrzu znajdują się wszystkie operacje/instrukcje wykonywane podczas działania algorytmu (za wyjątkiem instrukcji warunkowej). W tym bloku można zdefiniować zmienną, zmienić jej wartość, napisać słownie jakąś prostą komendę (np „dopisz literę na koniec aktualnego stringa”).

Funkcja: przechowywanie instrukcji, któe powinny się w danym kroku algorytmu wykonać

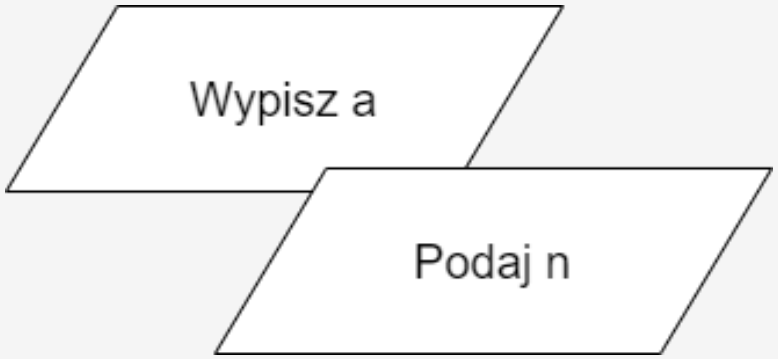
Schematy blokowe - symbole

- **=**, == to operatory porównania logicznego
- **:=**, =, <- to operatory przypisania wartości
- !=, **<>** to operatory nierówności dwóch wartości

Najczęściej stosuje się te pogrubione, jednakże spotkać można i schematy z innymi znakami (mogącymi zresztą być zależnymi od wybranego języka programowania) – warto więc być świadomym ich istnienia.

Schematy blokowe - symbole

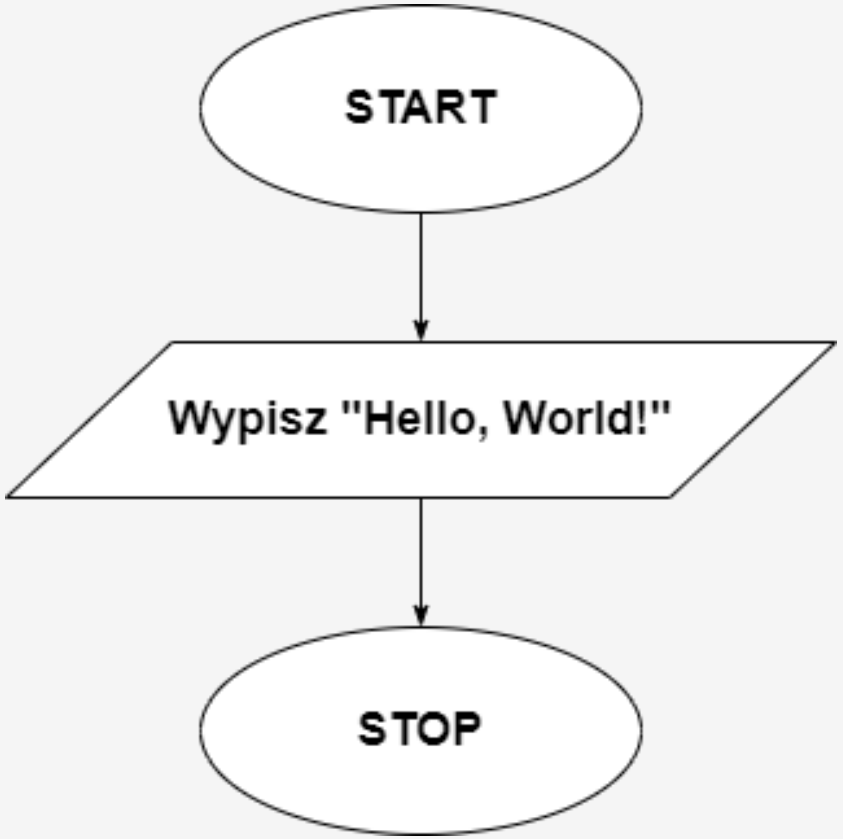
Skrzynka wejścia/wyjścia



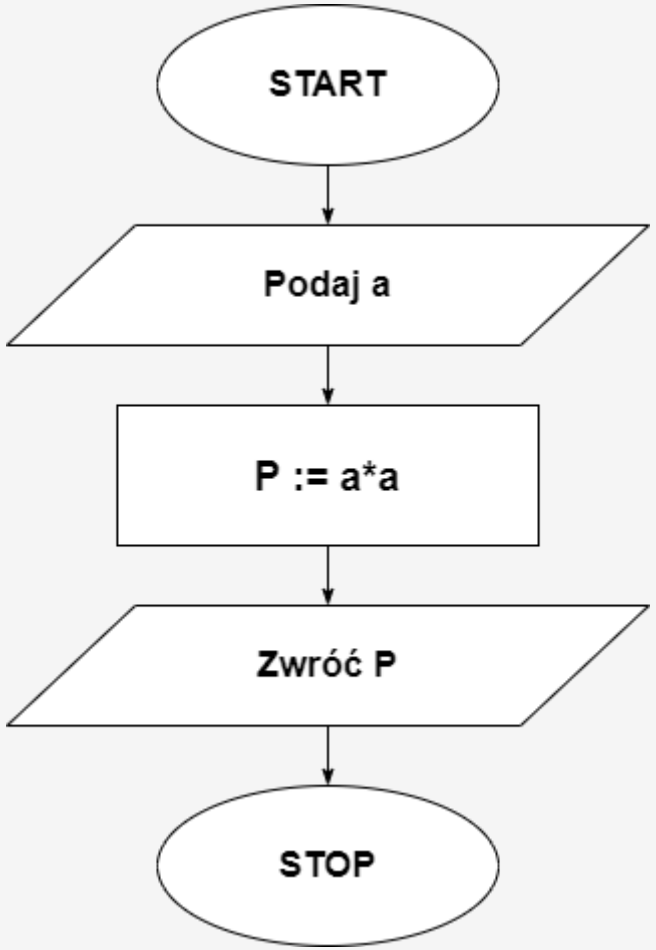
Reprezentowane na schemacie poprzez równoległobok, skrzynki wejścia/wyjścia służą do realizacji kontaktu z użytkownikiem poprzez komendy wypisujące wartości (np. print) bądź proszące użytkownika o podanie takowej (np. input).

Funkcja: wprowadzanie/wyprowadzanie danych

Algorytm Hello World



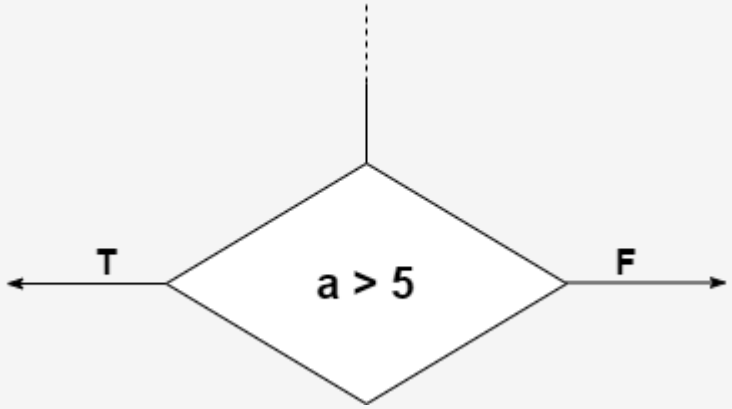
Obliczanie pola kwadratu



Schematy blokowe - przykład

Narysuj schemat blokowy dla algorytmu obliczającego pole trójkąta (wzór: $P = \frac{a*h}{2}$).

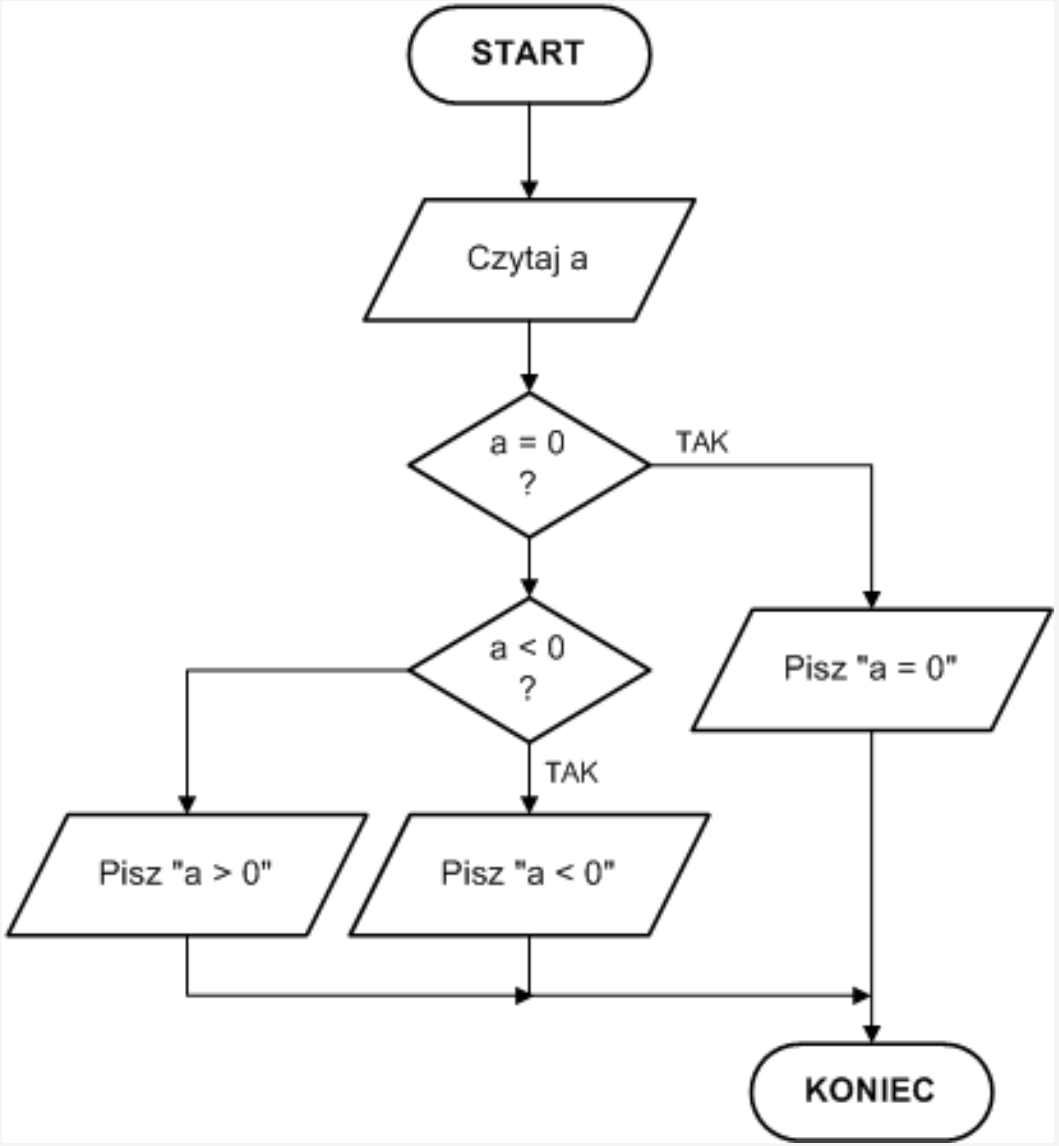
Skrzynka warunkowa



Skrzynka warunkowa przechowuje instrukcje wyboru (w rombie). Wchodzi do niej jedna strzałka, natomiast wychodzą dwie: dla wartości TAK (kiedy warunek zapisany w rombie jest prawdziwy) oraz dla wartości NIE (w momencie, którym nie jest).

Funkcja: sprawdzanie warunku (PRAWDA/FAŁSZ)

Schematy blokowe



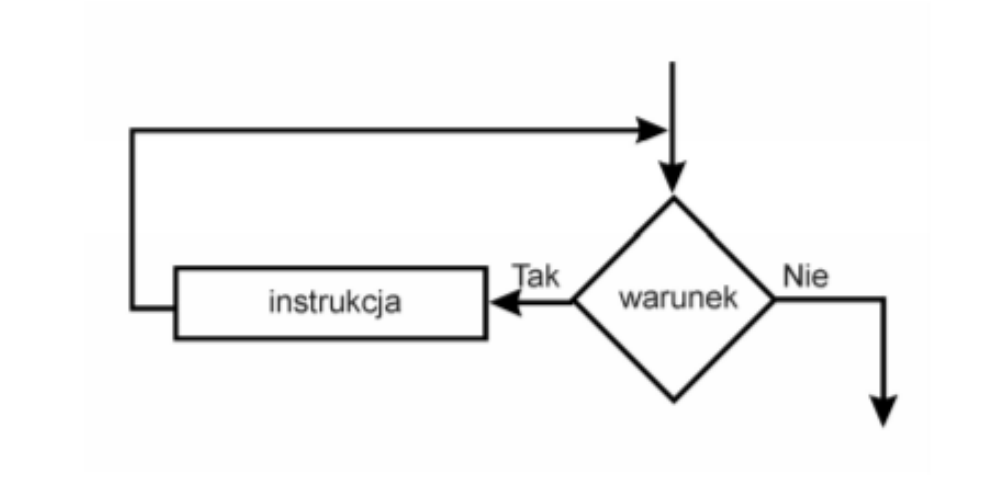
Schematy blokowe - przykład

Zaproponuj schemat blokowy algorytmu dzielącego jedną liczbę przez drugą (pamiętaj o ograniczeniach!)

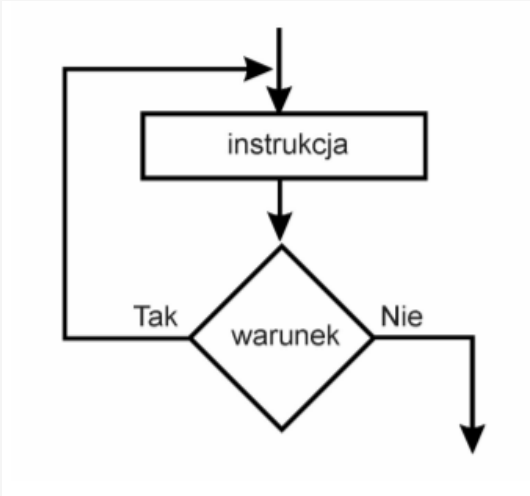
Schematy blokowe - przykłady

Algorytm: wczytuje dwie liczby. Rozpoznaje która jest większa i wypisuje w kolejności: większa, mniejsza. Jeżeli są równe wyświetla komunikat *Liczby są równe*.

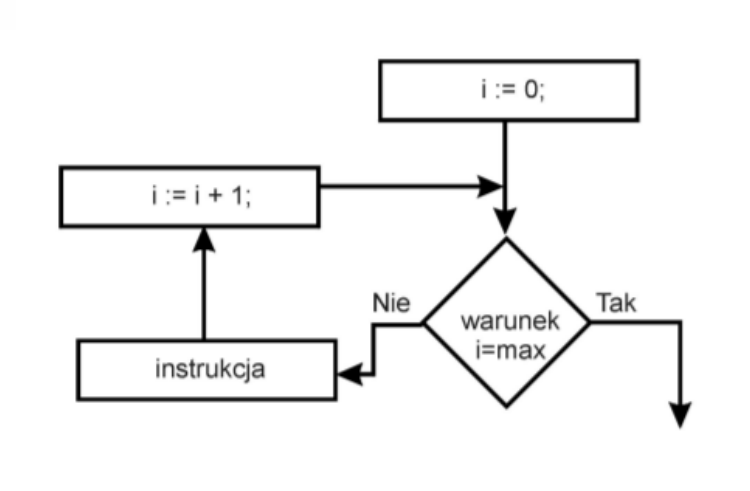
Pętla



Reprezentacja pętli **while**: najpierw sprawdzenie warunku, potem wykonanie instrukcji, ostatecznie wrócenie do punktu sprawdzenia warunku (rozpoczęcie kolejnej iteracji w pętli).

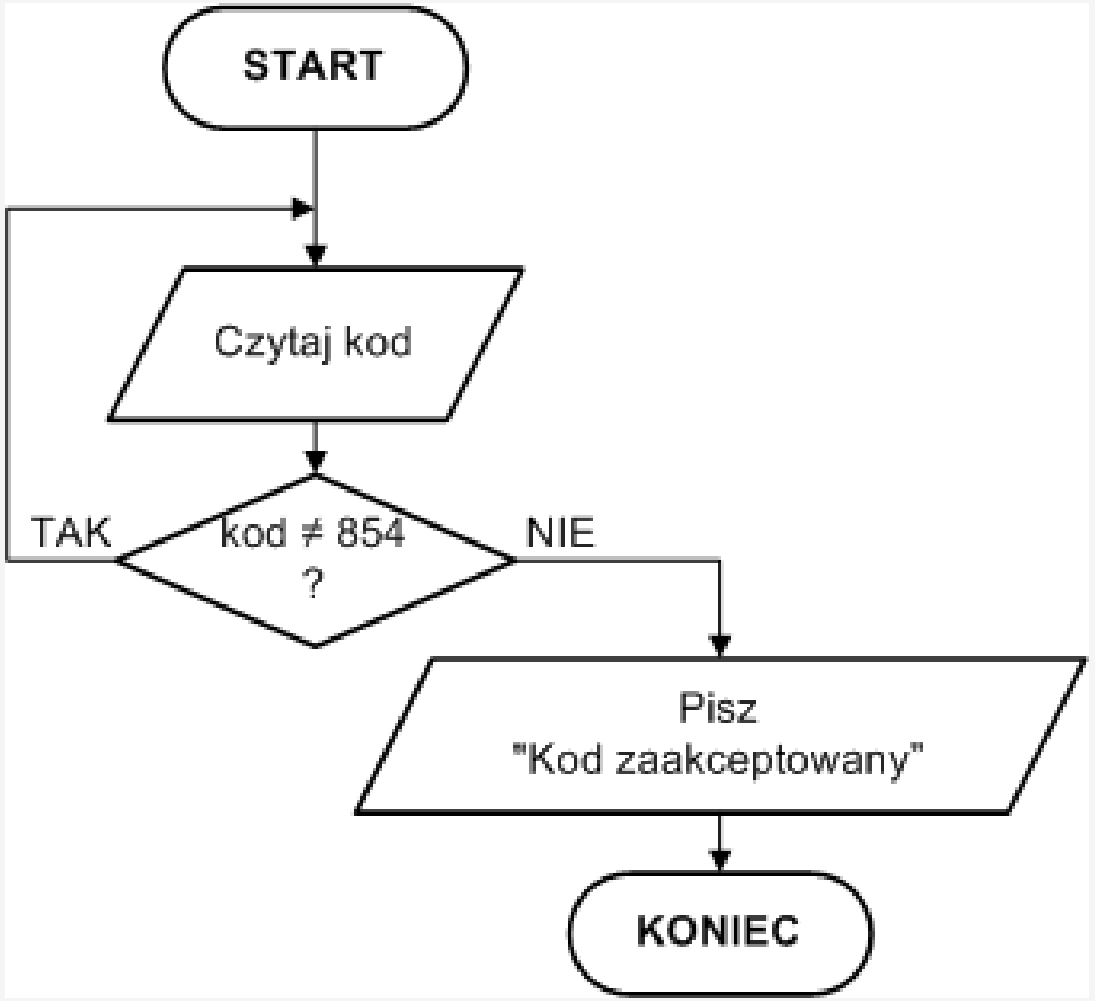


Reprezentacja pętli **until (do while)**: najpierw instrukcja, a potem sprawdzenie warunku (plus potencjalne powtórzenie wykonania instrukcji.

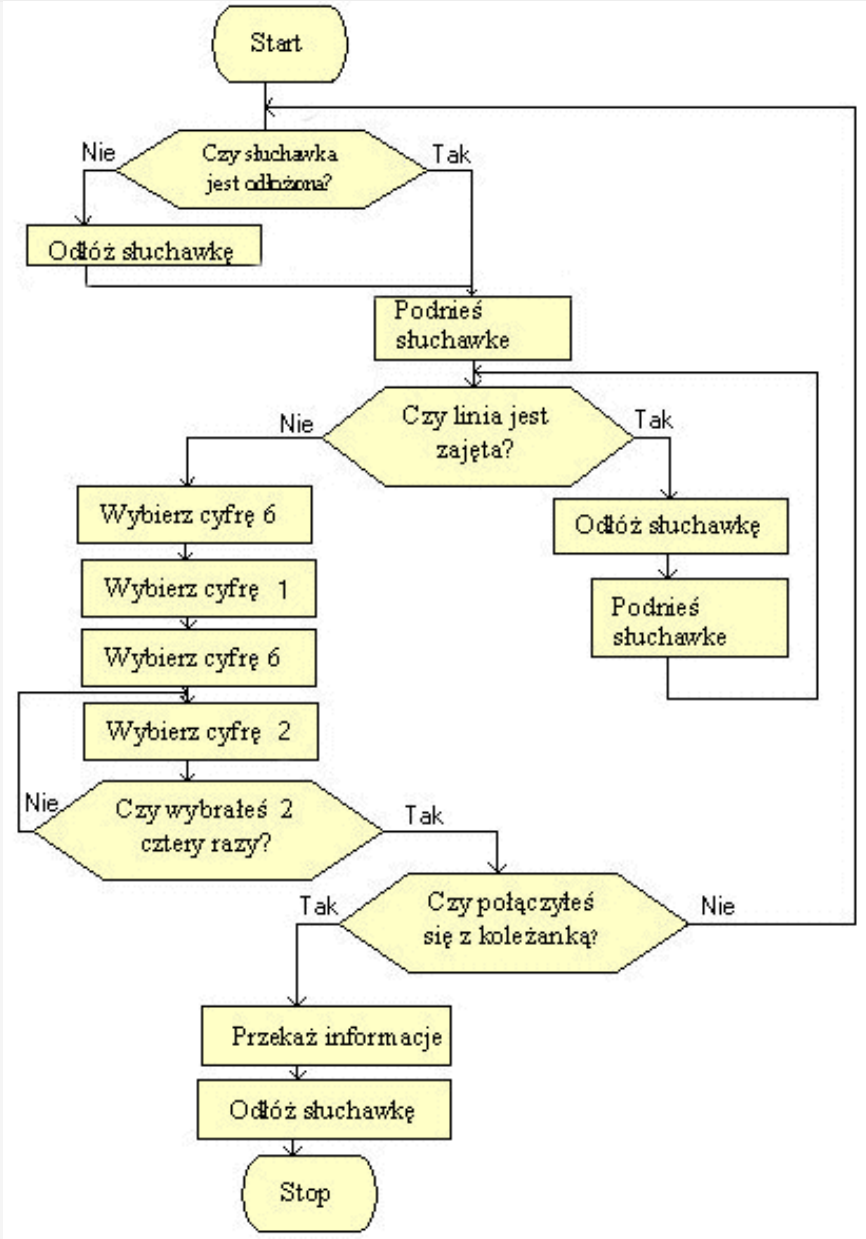


Symulacja pętli **for**: zapętlenie wykonywania się instrukcji określoną na schemacie ilość razy; *i* zwiększane do momentu zrównania się wartością z *max.*

Schematy blokowe



Schematy blokowe



Schematy blokowe - ćwiczenia

Przygotuj schemat blokowy dla algorytmu obliczającego sumę wszystkich liczb w liście podanej na wejściu.

Schematy blokowe - ćwiczenia

Zaproponuj schemat blokowy reprezentujący algorytm liczenia średniej arytmetycznej liczb w liście.

Schematy blokowe – sposób postępowania

1) Każda operacja jest umieszczona w skrzynce

2) Schemat ma tylko jedną skrzynkę „START" i przynajmniej jedną skrzynkę „KONIEC"

3) Skrzynki są ze sobą połączone.

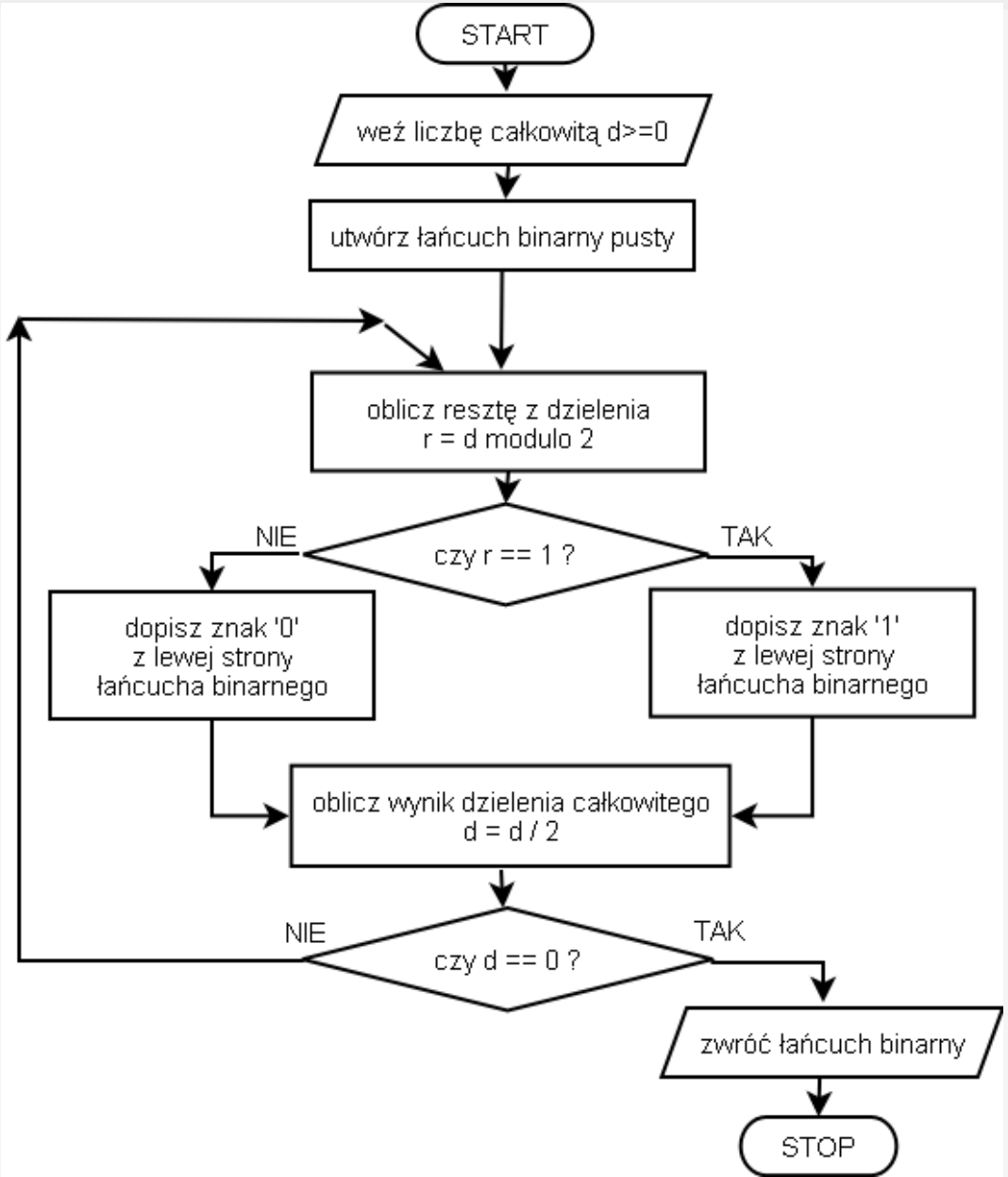
4) Ze skrzynki wychodzi jedno połączenie; wyjątek stanowią skrzynki: „KONIEC"
(z której nie wychodzą już żadne połączenia) oraz „warunkowa” (z której wychodzą dwa połączenia opisane TAK i NIE - w zależności od tego czy warunek jest spełniony, czy nie, można wyjść jedną z dwóch dróg)

5) Przepływ instrukcji może zostać zakłócony poprzez pętle lub skrzynki warunkowe

System dziesiętny -> system binarny

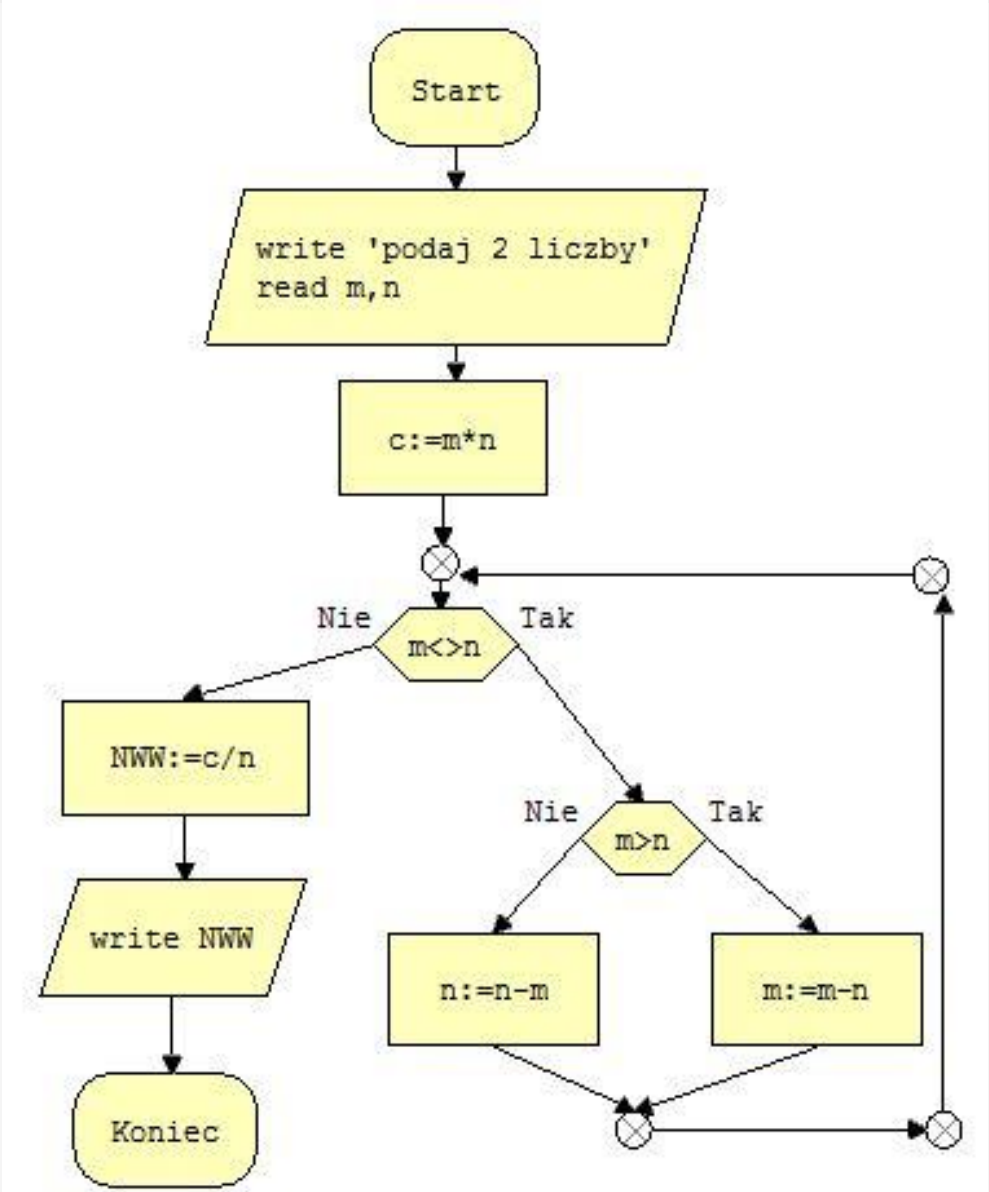
Napisz funkcję, która przyjmować będzie jeden argument: liczbę całkowitą w systemie dziesiętnym.
Zadaniem funkcji będzie zamiana reprezentacji tej liczby na system binarny (dwójkowy).
Funkcja powinna zwracać string, np. ’100110’.

System dziesiętny -> system binarny



Na podstawie schematu blokowego napisz funkcję wyliczającą najmniejszą wspólną wielokrotność dwóch liczb podanych jako argumenty.

Algorytmy


Start
write 'podaj 2 liczby'
read m,n
c:=m*n
Nie
m<>n
Tak
NWW:=c/n
Nie
m>n
Tak
write NWW
n:=n-m
m:=m-n
Koniec

Na podstawie schematu blokowego napisz funkcję wyliczającą największy wspólny dzielnik dwóch liczb podanych jako argumenty.

Algorytmy