Algorytm – ***opis działań*** *prowadzących do rozwiązania problemu lub osiągnięcia określonego celu*, **zawierający opis danych wejściowych oraz opis oczekiwanego wyniku** (***specyfikację problemu***).

Algorytm powinien być:

- **uniwersalny**: *zawsze dla poprawnych (zgodnych z opisem) danych powinien dawać poprawny wynik*

- **dobrze określony:** p*owinien łatwo dać się zapisać w postaci kodu w wybranym języku programowania*

Sposoby przedstawiana algorytmów

- Zapis słowny
- Schemat blokowy
- Pseudokod
- Kod programu

ZAPIS SŁOWNY

Język naturalny umożliwia zapis słowny algorytmu w formie logicznego ciągu kroków, które należy wykonać, aby osiągnąć pożądany efekt. Zaletą takiego sposobu zapisu jest prostota (nie trzeba zapamiętywać formalnych konstrukcji), do wad należą niedostateczna zwięzłość oraz możliwość błędnej interpretacji. W przypadku algorytmu Euklidesa taki słowny plan działań może przedstawiać się następująco:

1. Wybieramy dwie liczby naturalne.
2. Jeśli liczby są równe, to NWD jest np. pierwszą z nich i to oznacza koniec działań.
3. Jeśli liczby nie są równe, to trzeba:
   - zbadać, która jest większa;
   - odjąć od niej mniejszą i zastąpić większą przez otrzymaną różnicę;
   - wrócić do sprawdzenia warunku w punkcie 2 (pętla).

SCHEMAT BLOKOWY

Przedstawienie algorytmu w postaci graficznej z wykorzystaniem strzałek wskazujących kolejność wykonywania poszczególnych instrukcji umożliwiają schematy blokowe. Elementy blokowe, które stosuje się podczas tworzenia schematu, to:

– początek algorytmu (może być tylko jeden w schemacie);

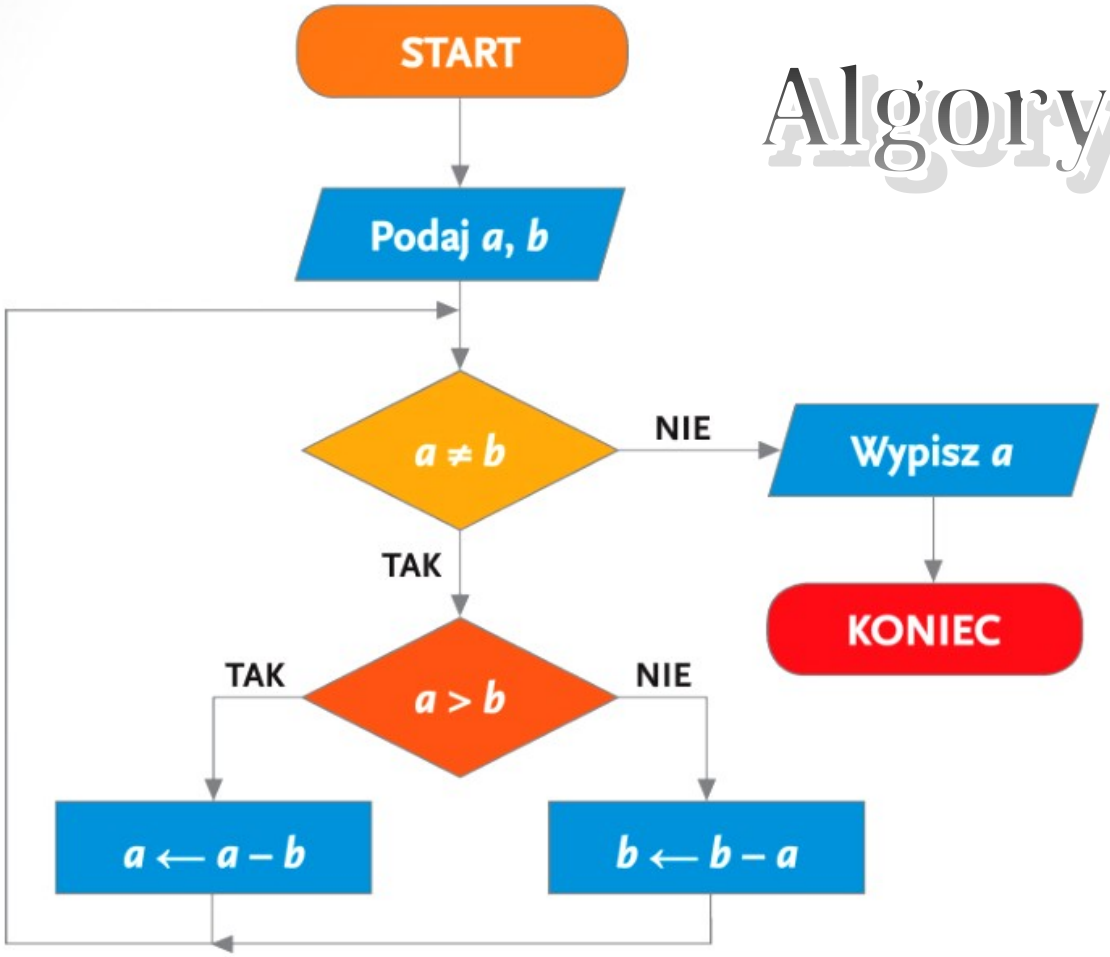
– operacje wejścia–wyjścia (wprowadzanie danych lub wyprowadzanie wyników);

– sprawdzanie warunku (TAK – kierunek, gdy warunek jest spełniony; NIE – kierunek, gdy warunek nie jest spełniony);

– działania wykonywane na danych;

– koniec algorytmu (może być ich kilka).

Algorytm Euklidesa



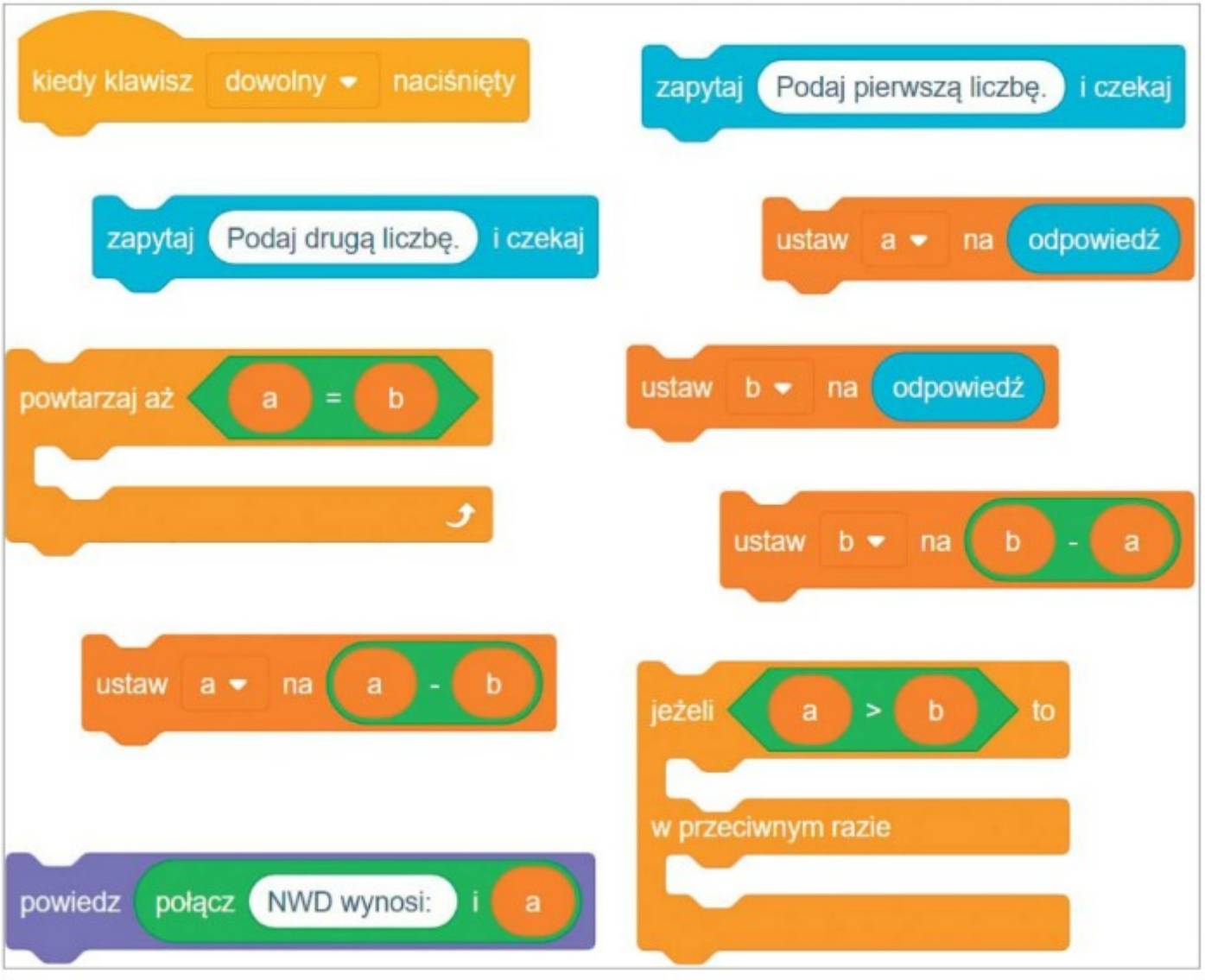
Schemat blokowy (wersja z odejmowaniem)

Algorytm Euklidesa – opisany około 300 lat p.n.e. ***opis działań pozwalających znaleźć największy wspólny dzielnik dwóch liczb całkowitych***. Autorstwo tego algorytmu przypisuje się greckiemu matematykowi – **Euklidesowi**. Algorytm ten jest najstarszym wykorzystywanym nadal algorytmem numerycznym.

PSEUDOKOD

Jeszcze innym sposobem prezentacji algorytmu jest pseudokod, czyli prosta i czytelna lista kroków zapisana za pomocą języka symbolicznego, korzystającego z języka naturalnego oraz operatorów (symboli dodawania, odejmowania itd.) i formuł matematycznych. Pseudokod nie ma ścisłych reguł zapisu; w przypadku algorytmu Euklidesa może wyglądać np. tak:

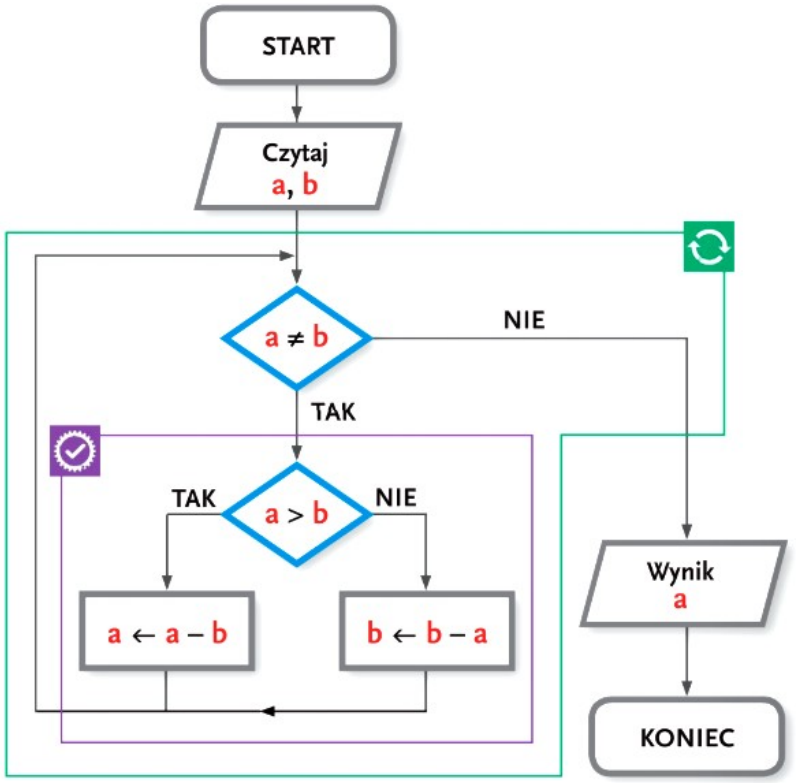
Wejście: 2 liczby naturalne, których NWD trzeba obliczyć.
Wyjście: liczba naturalna, NWD liczb podanych na wejściu.
Weź dwie liczby naturalne a i b.
Dopóki a różne od b, wykonuj:
Jeśli a większe od b, podstaw pod a wartość $a - b$;
w przeciwnym razie podstaw pod b wartość $b - a$.
Podaj a (lub b) jako wynik.


kiedy klawisz dowolny naciśnięty
zapytaj Podaj pierwszą liczbę. i czekaj
zapytaj Podaj drugą liczbę. i czekaj
ustaw a na odpowiedź
powtarzaj aż a = b
ustaw b na odpowiedź
ustaw b na b - a
ustaw a na a - b
jeżeli a > b to
w przeciwnym razie
powiedz połącz NWD wynosi: i a

Python	JavaScript

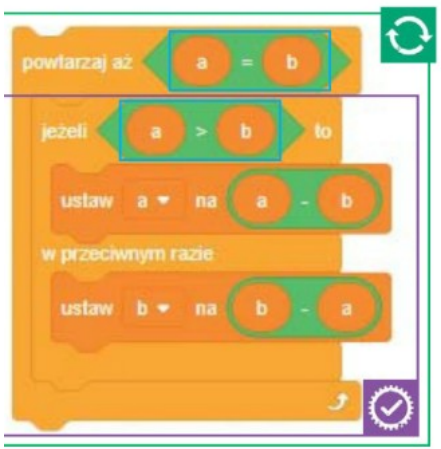
a = 152
b = 57
while a != b:
    if a > b:
        a = a - b
    else:
        b = b - a
print('NWD wynosi ' + str(a))
	var a = 152;
	var b = 57;
	while (a != b) {
	  if (a > b) {
	    a = a - b;}
	  else {
	    b = b - a;}
	}
	window.alert('NWD wynosi ' +
	↳String(a));

ZAPIS BLOKOWY

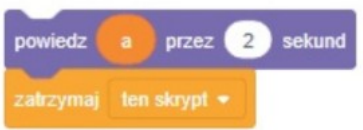



```
a = int(input("Podaj
pierwszą liczbę:"))

b = int(input("Podaj
drugą liczbę:"))
```




```
while a != b:
    if a > b:
        a = a - b
    else:
        b = b - a
```



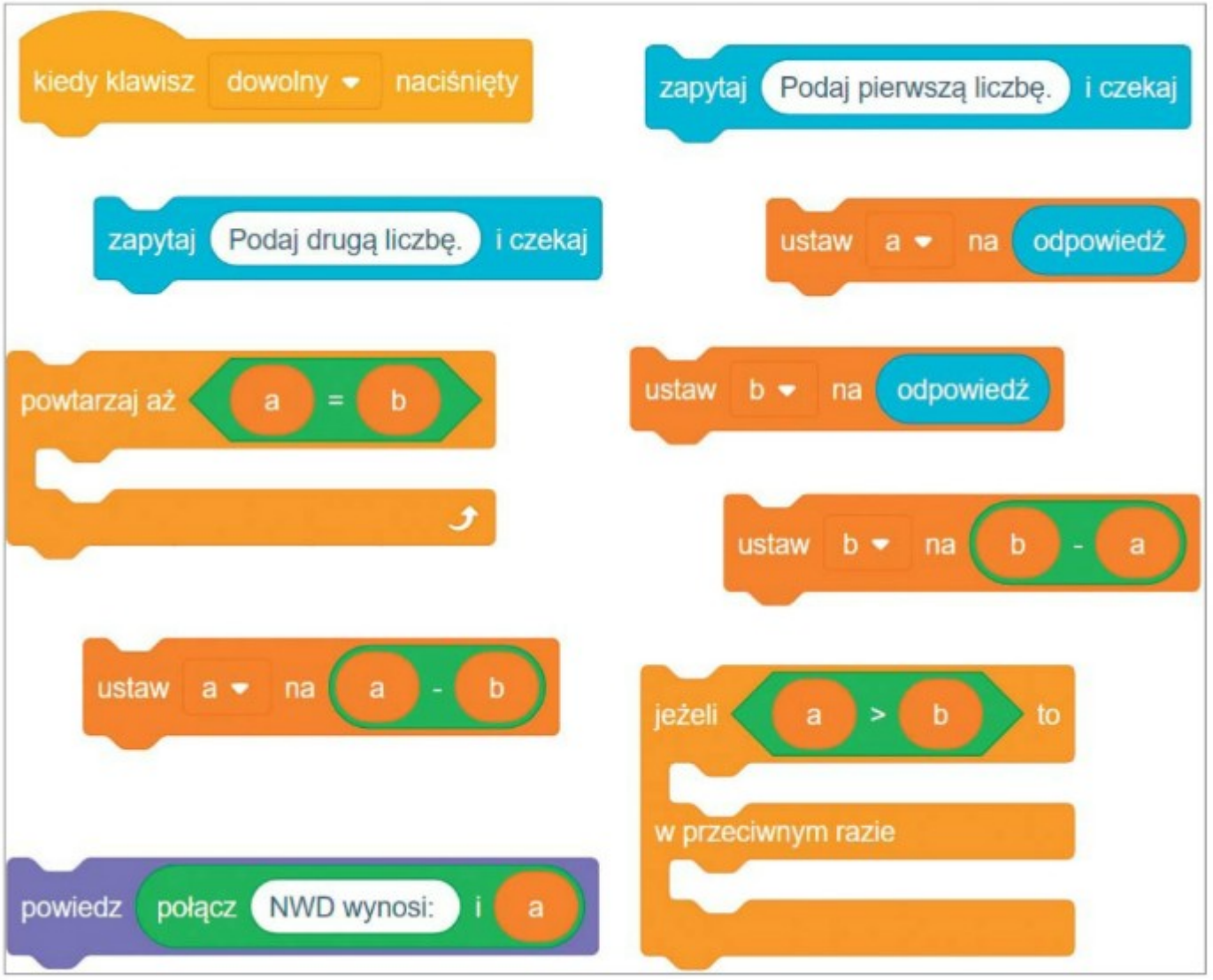

```
print (a)
```

Zmienna, np *a*, *b*, przechowuje wartość określonego typu, np. liczby całkowite.

Warunek to wyrażenie logiczne, np. *a* jest różne od *b* albo *a* jest większe od *b*. Może być spełniony lub nie.

Instrukcja warunkowa pozwala wybrać określoną akcję w zależności od tego, czy zdefiniowane wyrażenie logiczne jest prawdziwe, czy fałszywe.

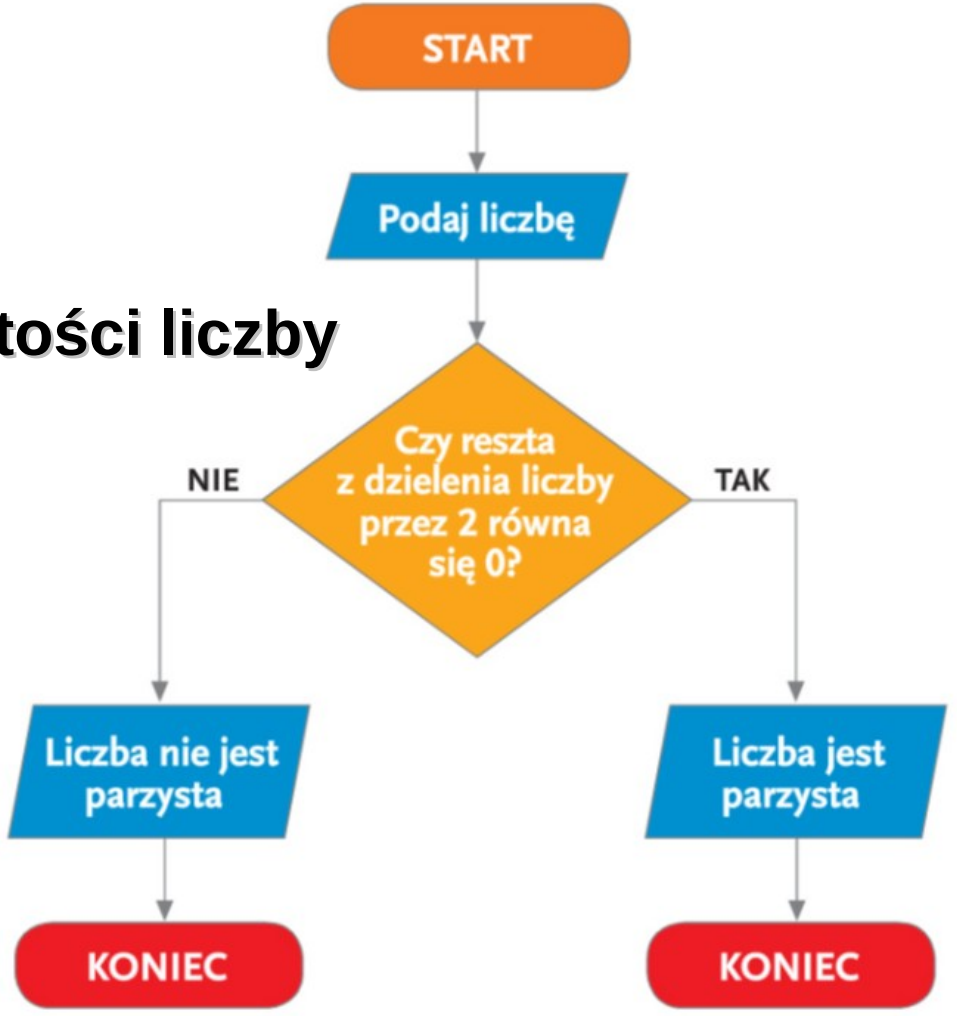
Pętle umożliwiają wielokrotne wykonywanie zadanych poleceń.


kiedy klawisz dowolny naciśnięty
zapytaj Podaj pierwszą liczbę. i czekaj
zapytaj Podaj drugą liczbę. i czekaj
ustaw a na odpowiedź
powtarzaj aż a = b
ustaw b na odpowiedź
ustaw b na b - a
ustaw a na a - b
jeżeli a > b to
w przeciwnym razie
powiedz połącz NWD wynosi: i a

Python	JavaScript

a = 152
b = 57
while a != b:
    if a > b:
        a = a - b
    else:
        b = b - a
print('NWD wynosi ' + str(a))
	var a = 152;
	var b = 57;
	while (a != b) {
	  if (a > b) {
	    a = a - b;}
	  else {
	    b = b - a;}
	}
	window.alert('NWD wynosi ' +
	↳String(a));

Badanie parzystości liczby

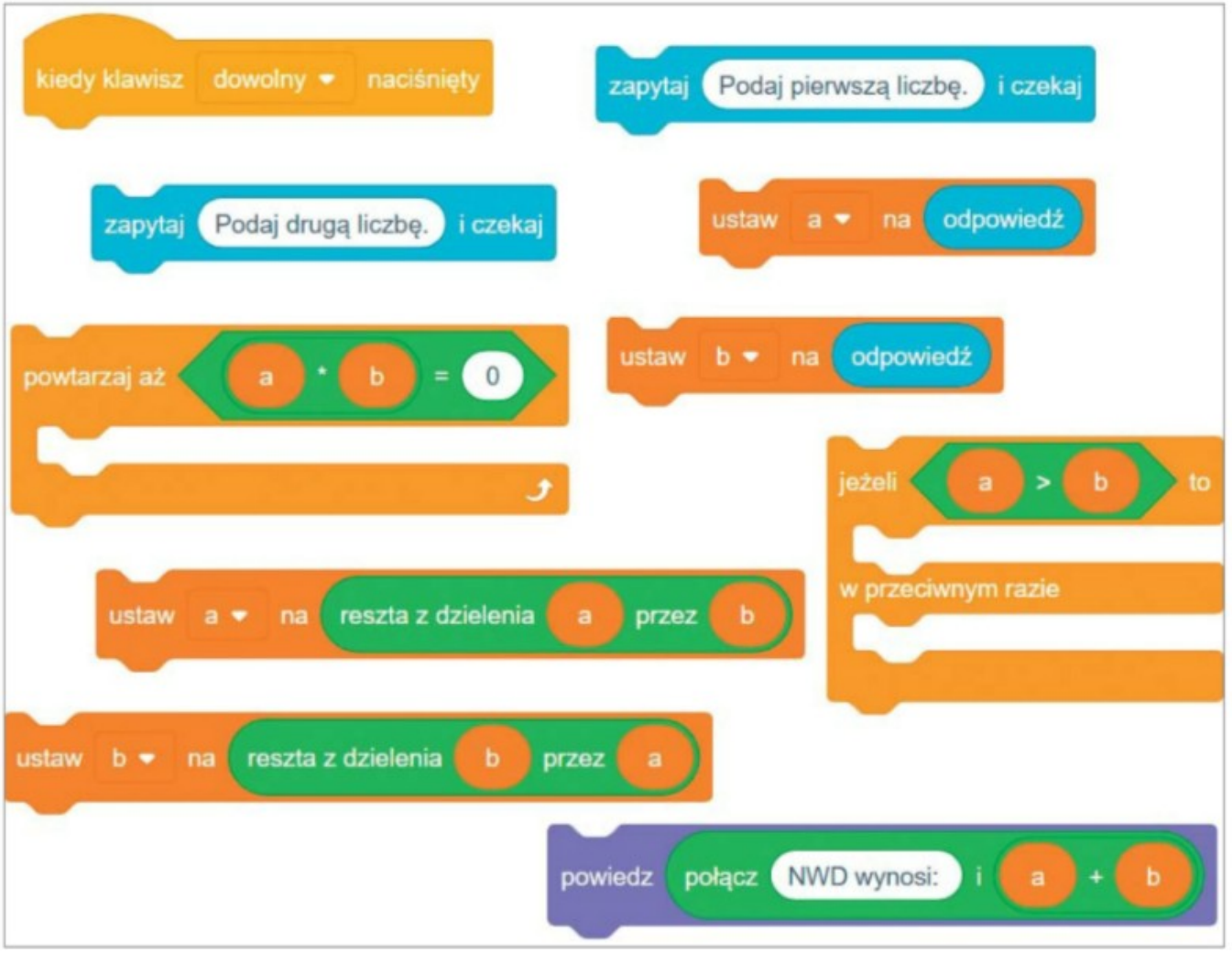


Badanie parzystości liczby

Euklides – wersja z dzieleniem

- Wybierane są dwie liczby naturalne.
- Dopóki żadna z liczb nie jest równa 0:
  - jeśli pierwsza liczba jest większa, to zastępuje się ją przez resztę z dzielenia całkowitego pierwszej liczby przez drugą;
  - w przeciwnym razie zastępuje się drugą liczbę przez resztę z dzielenia całkowitego drugiej liczby przez pierwszą (pętla);
- Jako wynik (NWD) podawana jest suma liczb (po zakończeniu pętli).

Euklides – wersja z dzieleniem



Na czym polega sortowanie?

Sortowanie – jeden z podstawowych problemów informatyki, polegający na **uporządkowaniu** zbioru danych **względem pewnych cech** charakterystycznych każdego elementu tego zbioru.
Szczególnym przypadkiem jest **sortowanie względem wartości** każdego elementu, np. sortowanie liczb, słów itp.

Algorytmy sortujące

- Sortowanie bąbelkowe
- Sortowanie przez wybór
- Sortowanie przez wstawianie
- Sortowanie przez scalanie
- Sortowanie szybkie
- Sortowanie kubełkowe

Sortowanie bąbelkowe *(bubble sort)*

Sortowanie bąbelkowe polega na porównywaniu dwóch kolejnych elementów zbioru i zamianie ich kolejności, jeżeli zaburza ona porządek, w jakim się sortuje zbiór (listę, tablicę)
Sortowanie **kończy się**, gdy podczas kolejnego przejścia **nie dokonano żadnej zmiany**

Sortowanie zbioru pięcioelementowego

Zbiór do posortowania: **[4, 2, 5, 1, 7]**

$$[\underbrace{4,2}_{4>2},5,1,7] \rightarrow [2,\underbrace{4,5}_{4<5},1,7] \rightarrow [2,4,\underbrace{5,1}_{5>1},7] \rightarrow [2,4,1,\underbrace{5,7}_{5<7}]$$

$$[\underbrace{2,4}_{2<4},1,5,7] \rightarrow [2,\underbrace{4,1}_{4>1},5,7] \rightarrow [2,1,\underbrace{4,5}_{4<5},7]$$

$$[\underbrace{2,1}_{2>1},4,5,7] \rightarrow [1,\underbrace{2,4}_{2<4},5,7]$$

$$[\underbrace{1,2}_{1<2},4,5,7]$$