Opis problemu i przedstawienie sposobu jego rozwiązania w postaci graficznej

Etapy rozwiązywania problemu

PROBLEM

1. sformułowanie zadania,
2. określenie danych wejściowych,
3. określenie celu czyli wyniku,

wybór metody rozwiązania

ALGORYTM

- opisu słownego,
- listy kroków,
- schematu blokowego,
- języka programowania

zapis algorytmu

PROGRAM

1. analiza poprawności rozwiązania
2. testowanie rozwiązania (programu) – ocena efektywności przyjętej metody

algorytm - przykład

Kolorowa BABKA

Sposób przyrządzania:

1. Masło utrzeć z cukrem.
2. Dodać żółtka, mąkę, 3 łyżki mleka i proszek do pieczenia.
3. Białka ubić na pianę, połączyć z ciastem.
4. Ciasto podzielić na 3 części.
5. Do jednej wsypać mak i wyłożyć do wysmarowanej tłuszczem i posypanej bułką tartą formy na babkę.
6. Do drugiej części dodać kakao, wyłożyć na ciasto z makiem.
7. Do trzeciej części dodać wiórki kokosowe, wyłożyć na wierzch.
8. Piec ok. 50 minut w temp. 160°C.
9. Czekoladę rozpuścić z pozostałym mlekiem i polać babkę.

Pojęcia

Algorytmika to nauka o algorytmach

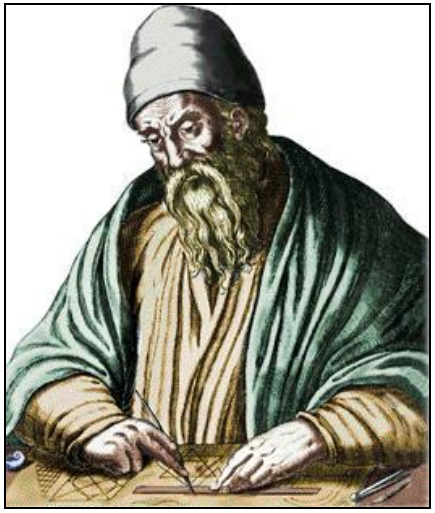
Algorytm jest to uporządkowany i uściślony sposób rozwiązania problemu, zawierający szczegółowy opis wykonywanych czynności.

Program komputerowy jest to logicznie uporządkowany ciąg instrukcji języka programowania, realizujący algorytm.

Język programowania jest to specjalny język, którym posługujemy się tylko przy pisaniu programu komputerowego. Każde słowo lub znak w tym języku jest instrukcja lub jego częścią.

Schemat blokowy jest to graficzne przedstawienie algorytmu.

Algorytm Euklidesa



Euklides (ur. ok. 365 p.n.e., zm. ok. 300 p.n.e.) – matematyk grecki pochodzący z Aten, przez większość życia działający w Aleksandrii.

Algorytm Euklidesa – algorytm znajdowania największego wspólnego dzielnika (NWD) dwóch różnych liczb naturalnych. Nie wymaga rozkładania liczb na czynniki pierwsze.

Opiera się on na fakcie, że jeśli od większej liczby odejmiemy mniejszą, to ta mniejsza liczba i otrzymana różnica będą miały taki sam największy wspólny dzielnik jak pierwotne liczby.
Gdy przy kolejnym odejmowaniu otrzymamy parę takich samych liczb, to znaleźliśmy NWD.

Źródło: wikipedia.pl

Przykład: Oblicz NWD liczb 32 i 12

Metoda 1: Rozkład liczb na czynniki pierwsze

$32 : 2 = 16$
$16 : 2 = 8$
$8 : 2 = 4$
$4 : 2 = 2$
$2 : 2 = 1$

$12 : 2 = 6$
$6 : 2 = 3$
$3 : 3 = 1$

$$NWD_{(32,12)} = 2 \cdot 2 = 4$$

Metoda 2: Przy pomocy algorytmu Euklidesa

$32 - 12 = 20$
$20 - 12 = 8$
$12 - 8 = 4$
$8 - 4 = 4$

NWD liczb 32 i 12 wynosi 4 gdyż zgodnie z założeniem algorytmu w kolejnym odejmowaniu otrzymaliśmy liczbę 4.

Algorytm – Sposób zapisywania

Algorytm można przedstawić w postaci:

- opisu słownego,
- listy kroków,
- schematu blokowego,
- języka programowania

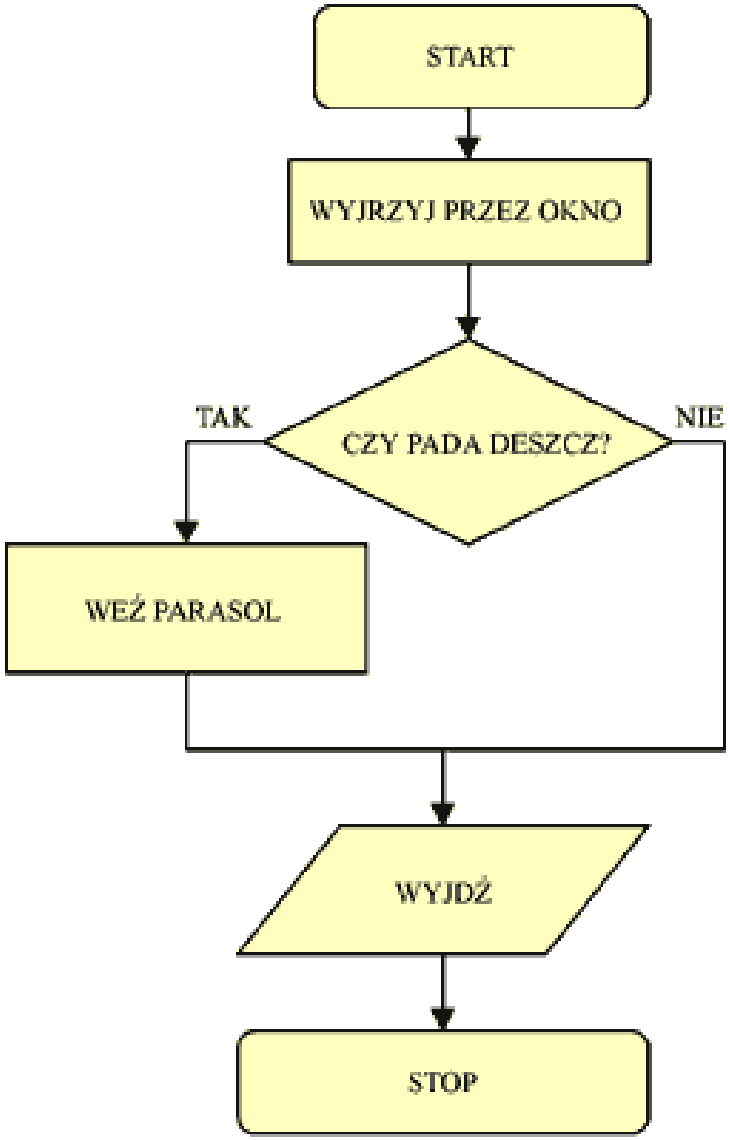
Lista kroków algorytmu - przykład

Kolorowa BABKA

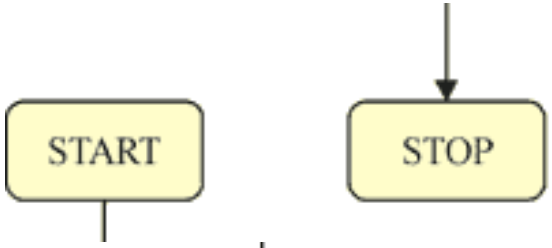
Sposób przyrządzania:

1. Masło utrzeć z cukrem.
2. Dodać żółtka, mąkę, 3 łyżki mleka i proszek do pieczenia.
3. Białka ubić na pianę, połączyć z ciastem.
4. Ciasto podzielić na 3 części.
5. Do jednej wsypać mak i wyłożyć do wysmarowanej tłuszczem i posypanej bułką tartą formy na babkę.
6. Do drugiej części dodać kakao, wyłożyć na ciasto z makiem.
7. Do trzeciej części dodać wiórki kokosowe, wyłożyć na wierzch.
8. Piec ok. 50 minut w temp. 160°C.
9. Czekoladę rozpuścić z pozostałym mlekiem i polać babkę.

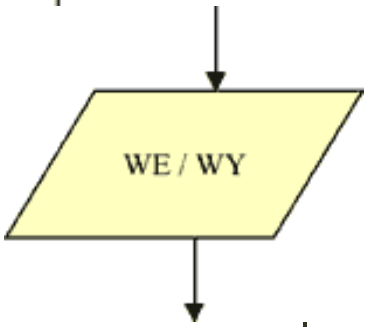
Schemat blokowy algorytmu



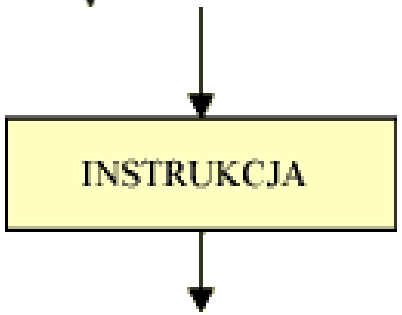
Elementy schematu blokowego



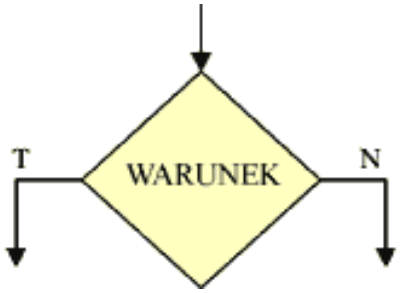
Skrzynka graniczna - wskazują początek i koniec każdego algorytmu



Skrzynka wejścia / wyjścia - umieszcza się wprowadzane dane lub wyprowadzane wyniki.



Skrzynka instrukcyjna - umieszcza się polecenia do wykonania (instrukcje) - podstawienie, obliczenie, wprowadzenie wartości



Skrzynka warunkowa - umieszcza się warunek, który decyduje o wyborze dalszej drogi postępowania.

Budowanie schematu blokowego

Zasady budowania schematu blokowego:

- każda operacja jest umieszczona w skrzynce,
- schemat ma tylko jedną skrzynkę „początek” i jedną skrzynkę „koniec”,
- skrzynki są ze sobą połączone,
- ze skrzynek wychodzi jedno połączenie z wyjątkiem skrzynek „początek” i „koniec” oraz „skrzynki warunkowej”

Lista kroków algorytmu

Ćwiczenie: **Przedstaw w postaci listy kroków algorytm obliczania pola trójkąta.**

Dane: **dowolne liczby rzeczywiste dodatnie a, h (a – długość boku trójkąta, h długość wysokości trójkąta).**

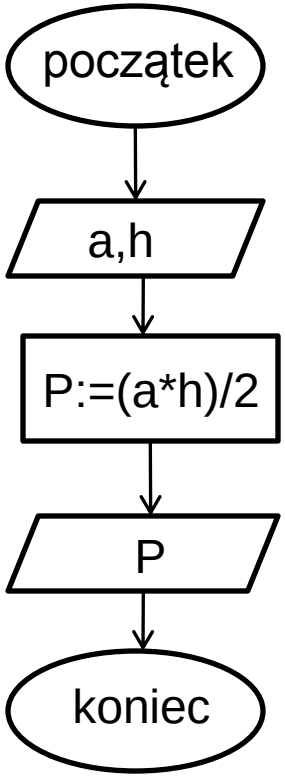
Wynik: **wartość pola trójkąta: P**

1. **Zacznij algorytm.**
2. **Wprowadź wartość boku a i wysokości h.**
3. **Zmiennej P przypisz wartość wyrażenia:** $(a*h)/2$
4. **Wyprowadź wynik: P.**
5. **Zakończ algorytm.**

Schemat blokowy algorytmu

Ćwiczenie: **Przedstaw w schematu blokowego algorytm obliczania pola trójkąta.**

1. **Zacznij algorytm.**
2. **Wprowadź wartość boku a i wysokości h.**
3. **Zmiennej P przypisz wartość wyrażenia:** $(a*h)/2$
4. **Wyprowadź wynik: P.**
5. **Zakończ algorytm.**



Lista kroków algorytmu

W skrzynce operacyjnej zamiast znaku = pojawia się oznaczenie := Taki zapis oznacza instrukcję przypisania (inaczej podstawienia); mówimy, że pod zmienną P podstawiona jest wartość wyrażenia (a*h)/2

Lista kroków algorytmu

Zadanie 1: **Zapisz specyfikację zadania i listę kroków algorytmu obliczania pola trapezu.**

Dane: **dowolne liczby rzeczywiste dodatnie a, b, h (a, b – długości podstaw trapezu, h długość wysokości trapezu).**

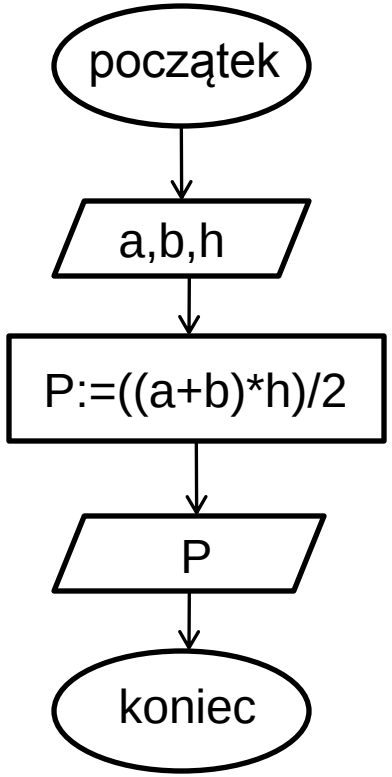
Wynik: **wartość pola trapezu: P**

1. **Zacznij algorytm.**
2. **Wprowadź wartość podstaw a, b i wysokości h.**
3. **Zmiennej P przypisz wartość wyrażenia:** $((a+b)*h)/2$
4. **Wyprowadź wynik: P.**
5. **Zakończ algorytm.**

Lista kroków algorytmu

Zadanie 1: Narysuj schemat blokowy algorytmu obliczania pola trapezu.

1. **Zacznij algorytm.**
2. **Wprowadź wartość podstaw *a, b* i wysokości *h*.**
3. **Zmiennej *P* przypisz wartość wyrażenia:** $((a+b)*h)/2$
4. **Wyprowadź wynik: *P*.**
5. **Zakończ algorytm.**



Praca domowa

Zadanie:

Dane są trzy liczby. Przedstaw graficznie algorytm obliczania ich sumy i średniej arytmetycznej

Jak realizować sytuacje warunkowe?

Algorytmika – sytuacja warunkowa

Z sytuacją warunkową mamy do czynienia wówczas, gdy wynik lub dalsze działanie zależy od spełnienia warunku.

Jeśli spełniony jest warunek W, wykonaj instrukcję A
lub
Jeśli spełniony jest warunek W, wykonaj instrukcję A, w przeciwnym wypadku wykonaj instrukcję B

Wpisując warunek logiczny stosujemy różne znaki relacji: „=” równy, „<>” różny, „<” mniejszy, „>” większy, „<=” mniejszy równy, „>=” większy równy. Można również wpisywać warunki złożone połączone spójnikami and (i) lub or (lub).

Algorytmika – sytuacja warunkowa

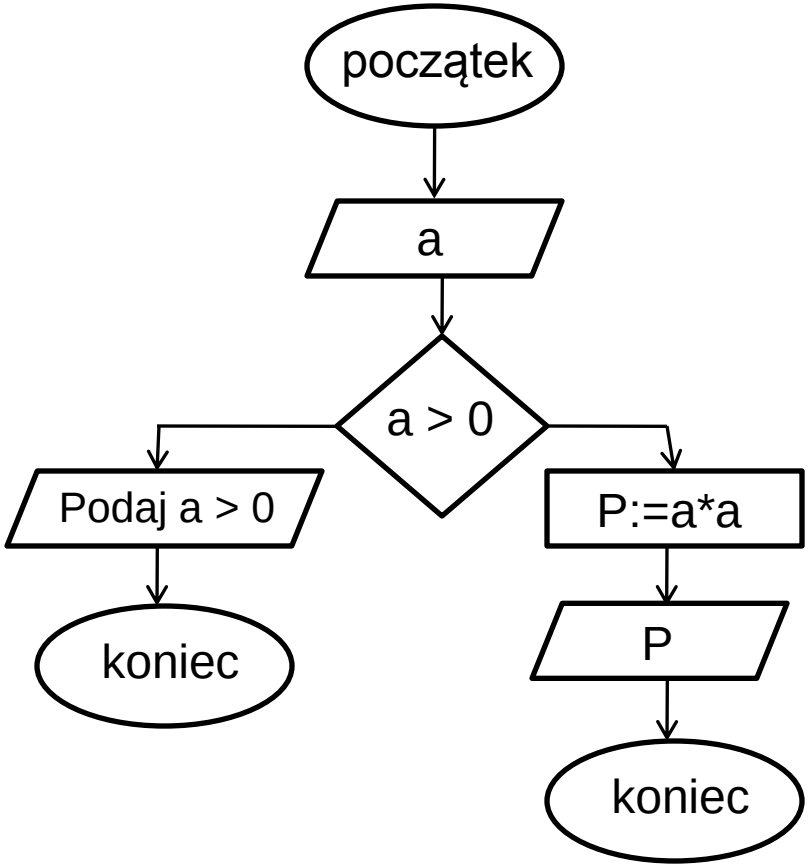
Zadanie:

Narysuj schemat blokowy pokazujący obliczanie pola P kwadratu o boku a.

Należy tu uwzględnić warunek: dla a>0 ma nastąpić obliczenie pola kwadratu i wyprowadzenie wyniku, w przeciwnym wypadku program ma wyświetlić odpowiedni komunikat i zakończyć działanie.

Algorytmika – sytuacja warunkowa

1. Zacznij algorytm.
2. Wprowadź wartość boku a.
3. Sprawdź warunek a>0
4. Jeżeli warunek nie jest spełniony wyświetl komunikat np. „Podaj a>0”
5. W przeciwnym wypadku za P podstaw wartość wyrażenia a*a
6. Wyprowadź wynik: P.
7. Zakończ algorytm.



Praca domowa

Zadanie:

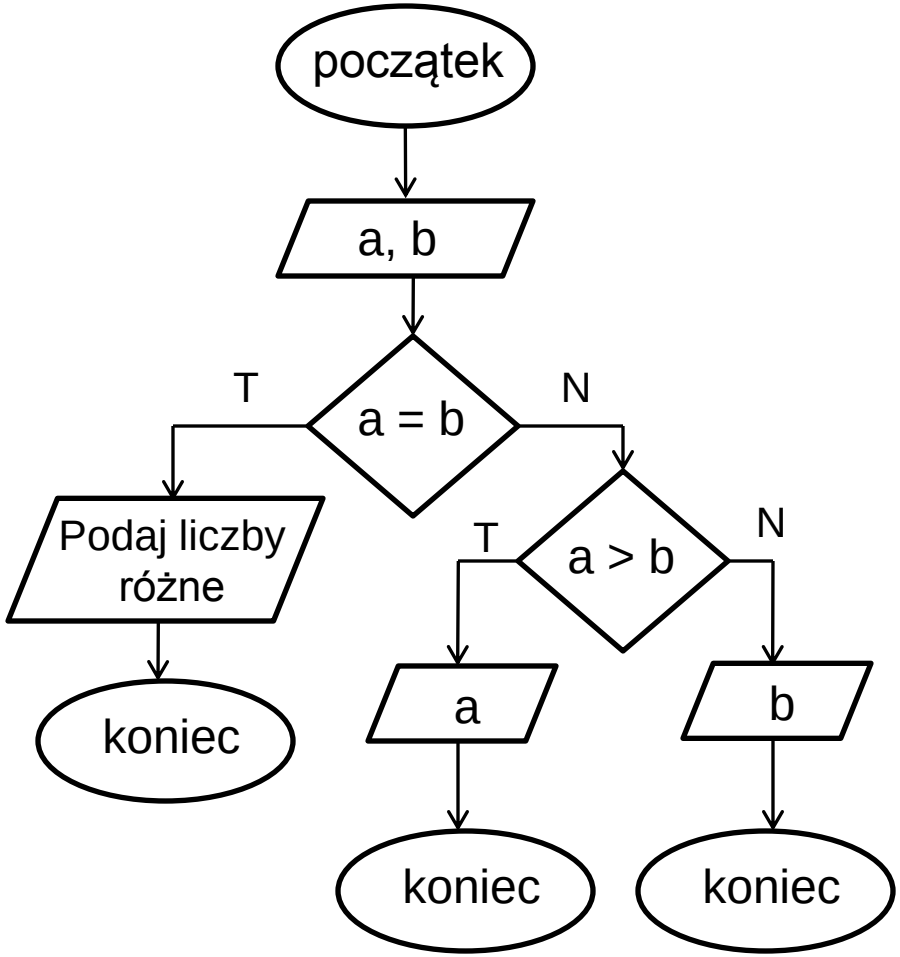
Wprowadź dwie liczby a i b. Przedstaw w postaci ciągu kroków i graficznie algorytm porównywania tych liczb i wyprowadzania na wyjściu większej z nich.

Zadanie:

Wprowadź dwie liczby a i b. Przedstaw w postaci ciągu kroków i graficznie algorytm porównywania tych liczb i wyprowadzania na wyjściu większej z nich.

Algorytmika – sytuacja warunkowa

1. Zacznij algorytm.
2. Wprowadź liczby a i b.
3. Sprawdź warunek $a = b$
4. Jeżeli warunek jest spełniony wyświetl komunikat np. „*Podaj liczby różne*”
5. W przeciwnym wypadku sprawdź warunek $a > b$
6. Jeśli warunek jest spełniony większa liczba to a
7. W przeciwnym wypadku liczba większa to b
8. Zakończ algorytm.



Na czym polega iteracja?

Napisz algorytm obliczania sumy siedmiu liczb.

1. Zacznij algorytm.
2. Suma jest równa zero.
3. **Wprowadź pierwszą liczbę.**
4. **Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.**
5. Wprowadź drugą liczbę.
6. Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.
7. Wprowadź trzecią liczbę.
8. Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.
9. Wprowadź czwartą liczbę.
10. Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.
11. Wprowadź piątą liczbę.
12. Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.
13. Wprowadź szóstą liczbę.
14. Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.
15. Wprowadź siódmą liczbę.
16. Do poprzedniego wyniku sumy dodaj wprowadzoną liczbę.
17. Wyprowadź wynik sumy.
18. Zakończ algorytm.

Algorytmika - iteracja

Zadanie: Oblicz sumę siedmiu liczb.

Dane: Siedem dowolnych liczb rzeczywistych, kolejno zapamiętywanych w zmiennej a.

Wynik: wartość sumy: $Suma$

1. Zacznij algorytm.
2. Zmiennej $Suma$ przypisz wartość zero: $Suma := 0$.
3. Wprowadź liczbę i zapamiętaj ją w zmiennej a.
4. Do wyniku sumy dodaj wprowadzoną liczbę: $Suma := Suma + a$.
5. Jeśli nie jest to siódma liczba, wróć do kroku 3.
6. Wyprowadź wynik: $Suma$
7. Zakończ algorytm.

Algorytmika – Iteracja

Iteracja to technika algorytmiczna polegająca na wykonaniu tej samej instrukcji dla n zmiennych (np. liczb, wartości logicznych).

Stosujemy wtedy tzw. **pętlę**. Z pętlą mamy do czynienia, gdy w pewnym kroku algorytmu wracamy do jednego z wcześniejszych kroków, co powoduje, że kroki te mogą zostać wykonane wiele razy.

W algorytmach iteracyjnych liczba wykonanych powtórzeń może być określona przez ilość wprowadzonych liczb lub zależeć od warunku.

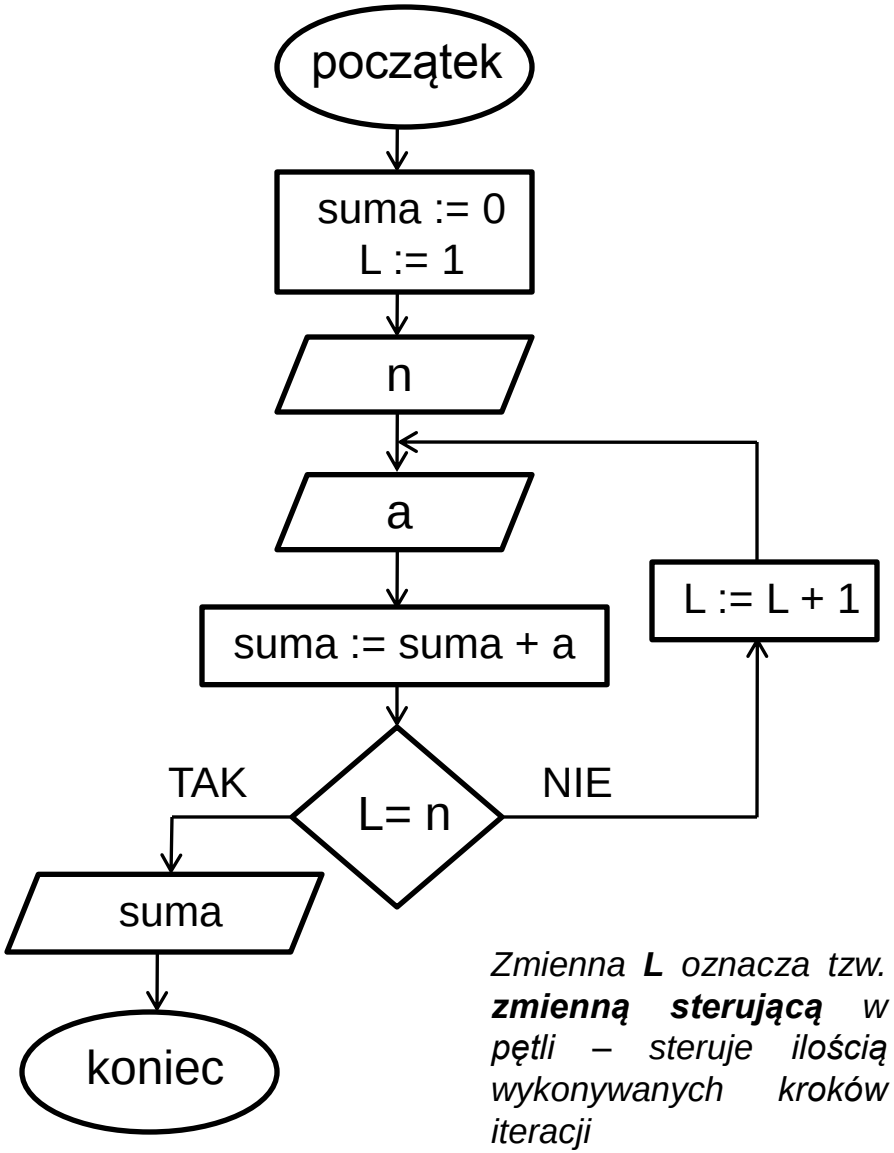
Algorytmika - iteracja

Zadanie: Oblicz sumę n liczb rzeczywistych.

Dane: Liczba naturalna n różna od zera, oznaczająca ilość wprowadzanych liczb, n dowolnych liczb rzeczywistych, z których każda kolejno zapamiętywana jest w zmiennej a.

Wynik: wartość sumy: $Suma$

Algorytmika – iteracja



*Zmienna **L** oznacza tzw. **zmienną sterującą** w pętli – steruje ilością wykonywanych kroków iteracji*

Suma := 0 **a – 7, 20, 13, 15**
L := 1
n := 4

krok 1
a := 7
suma := 7
czy L = n – NIE
L := 2

krok 2
a := 20
suma := 27
czy L = n – NIE
L := 3

krok 3
a := 13
suma := 40
czy L = n – NIE
L := 4

krok 4
a := 15
suma := 55
czy L = n – TAK

Wyprowadzenie liczby 55

Na czym polega programowanie?

Programowanie jest to przedstawienie algorytmu w postaci instrukcji języka programowania, zapisanych w kolejności wyznaczonych przez ten algorytm

Programowanie można podzielić na kilka etapów:

1. **Zapisanie algorytmu w postaci instrukcji języka programowania (kod źródłowy).**
2. **Translacja kodu źródłowego na plik wykonywalny.**
3. **Uruchomienie i wykonanie programu**

Kod źródłowy programu w języku Turbo Pascal

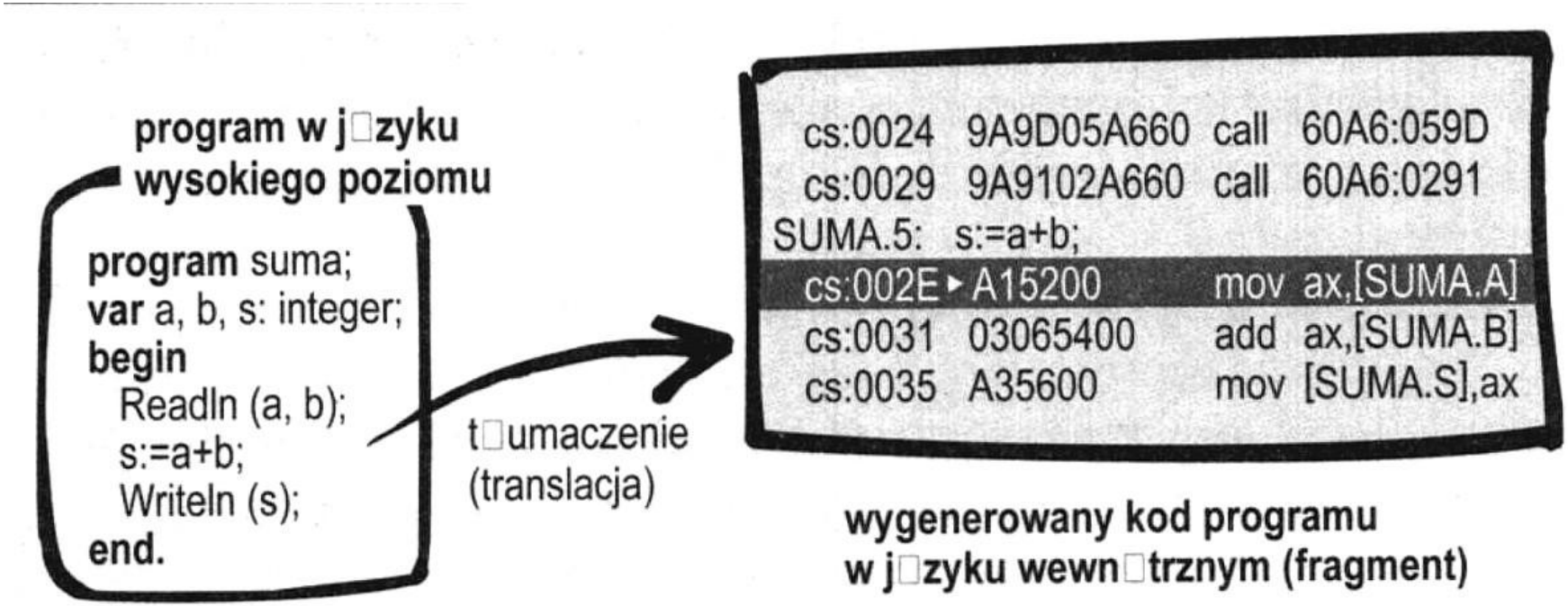
```
program p1;
uses crt;
var a,b,suma,iloczyn:integer;
begin
clrscr;
     write('Podaj pierwszą liczbę: ');
     readln(a);
     write('Podaj drugą liczbę: ');
     readln(b);
         suma:=a+b;
         iloczyn:=a*b;
     writeln('Suma liczb wynosi: ',suma, ' , a iloczyn:
',iloczyn);
end.
```

Translacja jest to tłumaczenie programu na język wewnętrzny komputera. Może ona przebiegać w formie kompilacji i interpretacji.

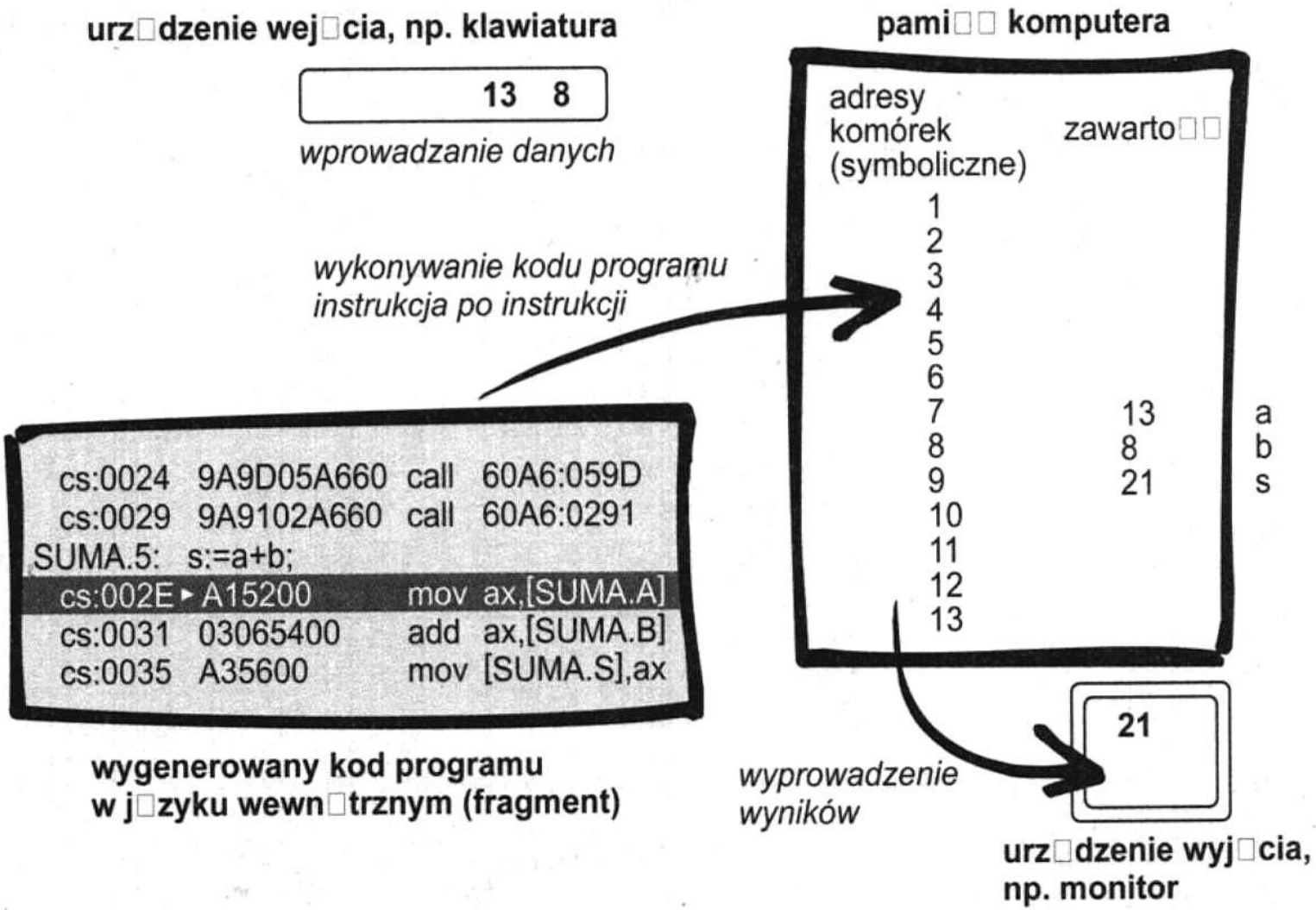
Kompilacja jest tłumaczeniem programu w całości i raz skompilowany program nie wymaga już powtórnej operacji tłumaczenia.

Interpretacja jest tłumaczeniem programu instrukcja po instrukcji przy każdorazowym uruchomieniu programu.

Tłumaczenie programu zapisanego w języku wysokiego poziomu na kod programu w języku wewnętrznym



Uruchomienie i wykonanie programu



Algorytmy sortowania

Algorytmika – Sortowanie

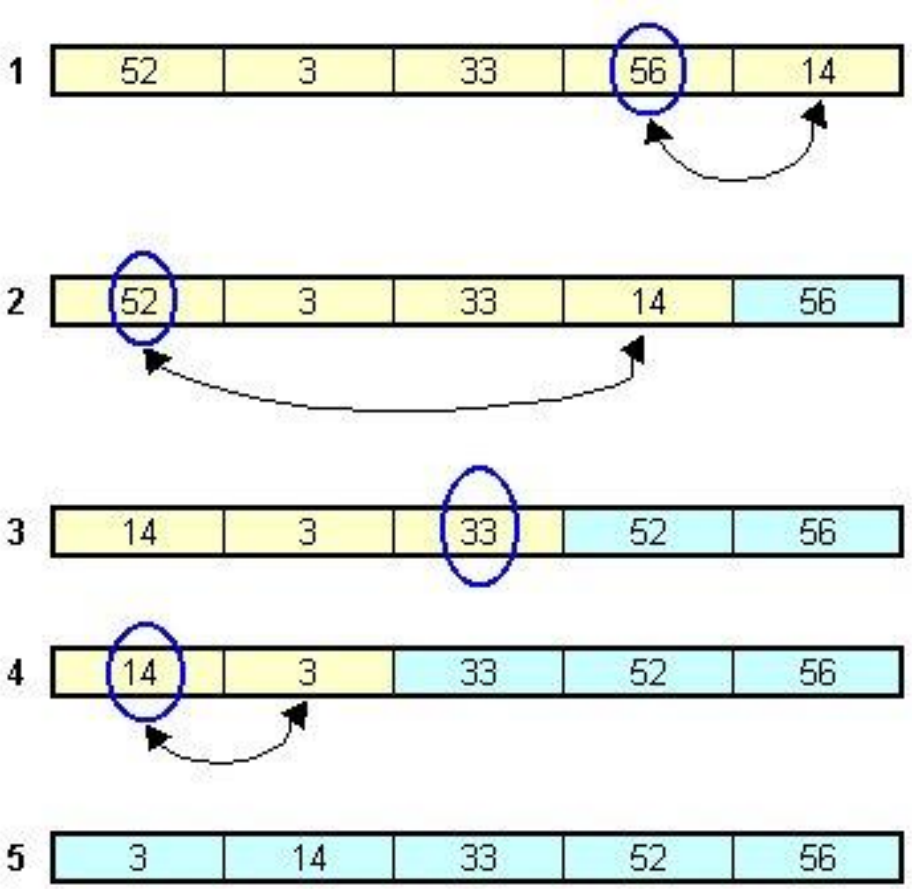
Wśród technik algorytmicznych ważne miejsce zajmują algorytmy sortowania czyli porządkowanie ciągu elementów np. liczb.

W algorytmice stosujemy następujące metody sortowania:

1. **Sortowanie przez wybór.**
2. **Sortowanie bąbelkowe**
3. **Sortowanie kubełkowe**

Algorytmika – Sortowanie przez wybór

Sortowanie przez wybór (np. od najmniejszej do największej) polega na wyszukaniu największej liczby, przestawieniu jej na końcu ciągu liczb i takim samym postępowaniu w ciągu z pominięciem pierwszego elementu.

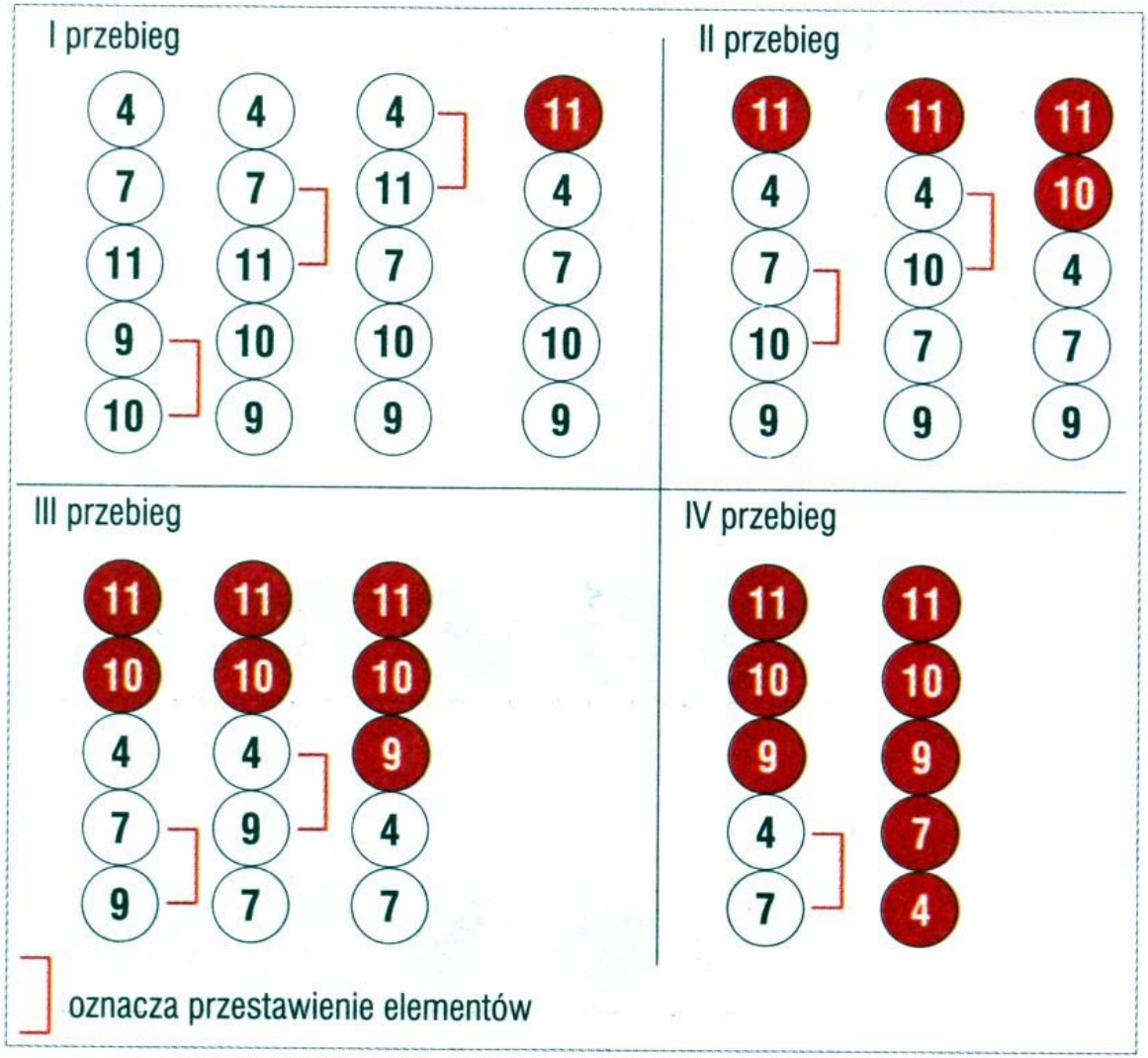


Algorytmika – Sortowanie przez wybór

```
program sortowanie_przez_wybor;
  uses crt;
  const n=10;
var
  tab: array[1..n] of integer;
  i,j,k,min:integer;
begin
  clrscr;
  writeln('Sortowanie przez wybor');
    for i:=1 to n do readln(tab[i]);
  writeln('Liczby przed sortowaniem:');
    for i:=1 to n do write(tab[i]);
    for j:=1 to n-1 do
       begin
         min:=j;
         for i:= j+1 to n do
         if tab[i] < tab[min] then min:= i;
         k:=tab[min];
           tab[min]:=tab[j];
           tab[j]:=k;
       end;
  writeln('Liczby po sortowaniu:');
    for i:=1 to n do write(tab[i]);
    repeat until
       keypressed;
end.
```

Algorytmika – Sortowanie bąbelkowe

Sortowanie bąbelkowe polega na porównywaniu parami kolejnych liczb i przestawianiu, jeśli są ustawione w niewłaściwej kolejności.

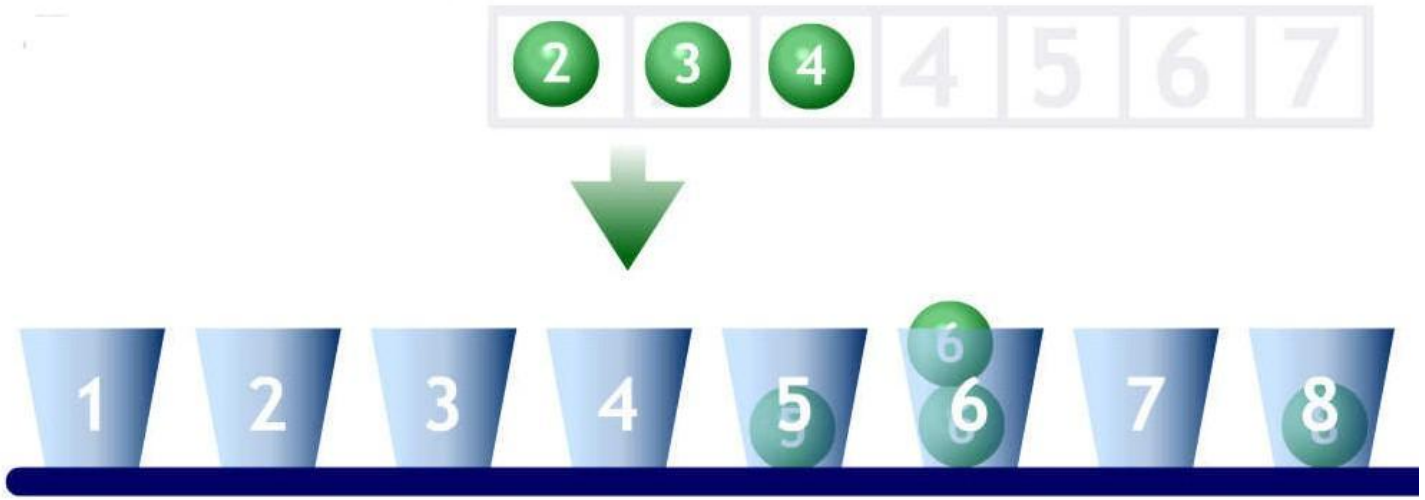


Algorytmika – Sortowanie bąbelkowe

```
program sortowanie_babelkowe;
uses crt;const n=10;
var  tab:array[1..n] of integer;
i,j,k: integer;
begin
clrscr;
writeln('Sortowanie babelkowe');
writeln('Wprowadzanie liczb');
for i := 1 to n do
readln(tab[i]);
writeln('Liczby przed sortowaniem:');
for i := 1 to n do write(tab[i]);
for j := 1 to n - 1 do
for i := 1 to n - 1 do
if tab[i] > tab[i+1] then
begin        k:= tab[i];
tab[i]:=tab[i+1];
tab[i+1]:=k;
end;
writeln('Liczby po sortowaniu:');
for i := 1 to n do
write(tab[i]);
repeat until
keypressed;
end.
```

Algorytmika – Sortowanie kubełkowe

Sortowanie kubełkowe polega na porównywaniu liter umieszczonych na tych samych pozycjach, począwszy od ostatniej litery najdłuższych słów. litera na danej pozycji decyduje o umieszczeniu słowa w ciągu w odpowiednim miejscu.



Algorytmika – Sortowanie kubełkowe

	mak rak			akt
a	k	m	r	t

mak rak	akt			
a	k	m	r	t

akt		mak	rak	
a	k	m	r	t

Algorytm sortowania kubełkowego często jest stosowany do alfabetycznego katalogowania wyrazów.

W celu posortowania takiego zbioru, w pierwszym kroku należy odnaleźć najdłuższe wyrazy i porównać ich ostatnie litery. Dzięki temu porównaniu możemy ustalić ich kolejność. W ten sposób porównujemy kolejne litery – zawsze istotna jest litera z określonej pozycji – dla nowo porównywanych ostatnia.

Uporządkujmy zbiór 3 wyrazów: rak, akt, mak. W zbiorze występuje pięć różnych liter – tyle potrzebujemy kubełków.

Kolej na opróżnienie kubełków - dokonujemy tego zaczynając od MAK AKT lewej od dołu - otrzymamy ciąg:
MAK, RAK, AKT
Ponownie „napełniamy kubełki”

Po opróżnieniu kubełków kolejność jest identyczna jak poprzednio, przejdźmy zatem do ostatniego etapu (według pierwszych liter).

Opróżnienie kubełków ustanawia ostateczną kolejność.
Posortowany ciąg to:
AKT MAK RAK

Na czym polega rekurencja?

Algorytmika – Rekurencja

O rekurencji mówimy wtedy gdy definiując pojęcie odwołujemy się do niego samego.