

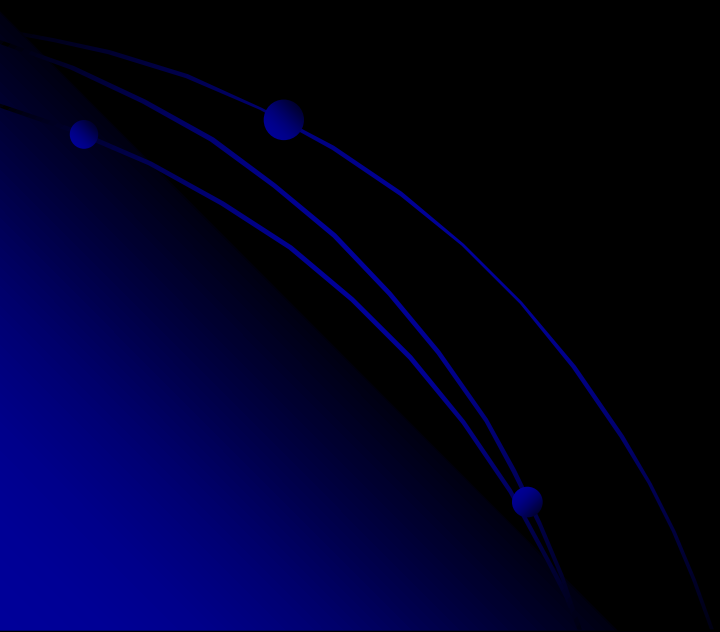
Algorytmika i programowanie

dr inż. Barbara Fryc

Wyższa Szkoła Informatyki i Zarządzania
w Rzeszowie

Algorytmy i sposoby ich przedstawiania

- Algorytm - informatyczny opis planu rozwiązania zadania
- Sposoby przedstawiania algorytmów



Algorytm - informatyczny opis planu rozwiązania

Pojęcie algorytmu

Wg Macieja M. Sysło ([8], s. 20).

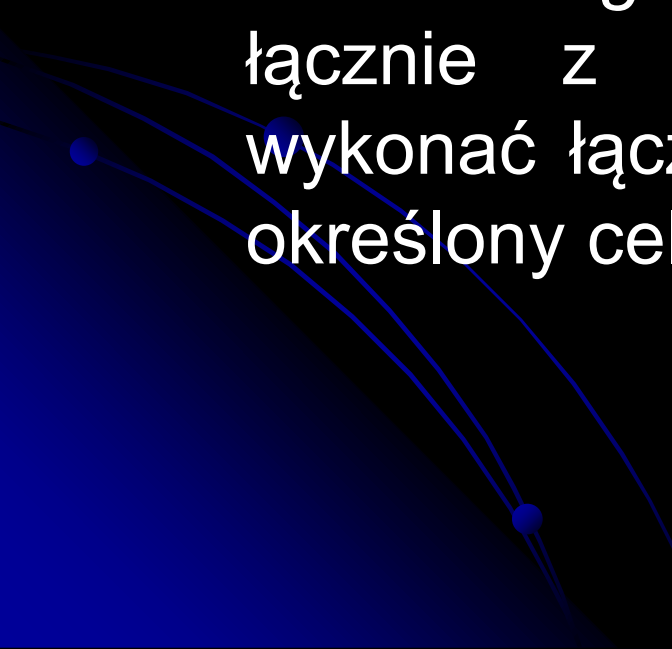
Algorytm jest przepisem rozwiązywania postawionego zadania, będącym dokładnie określonym układem elementarnych instrukcji wraz z porządkiem ich wykonania.

Algorytm - informatyczny opis planu rozwiązania

Pojęcie algorytmu

Wg Władysława Turskiego ([9], s. 59)

Przez *algorytm* należy rozumieć opis obiektów łącznie z opisem czynności, które należy wykonać łącznie z tymi obiektami, aby osiągnąć określony cel.



Algorytm - informatyczny opis planu rozwiązania

Pojęcie algorytmu

Wg Stefana Węgrzyna ([10], s. 12)

Algorytm jest przepisem opisującym krok po kroku rozwiązanie problemu lub osiągnięcie jakiegoś celu.



Algorytm - informatyczny opis planu rozwiązania

Pojęcie algorytmiki

Algorytmika jest nazwą dziedziny zajmującej się algorytmami i ich własnościami, projektowaniem i analizą algorytmów.

(Dawid Harel: „Rzecz o istocie informatyki - algorytmika”).

Pochodzenie terminu „algorytm”

Termin algorytm pochodzi od brzmienia fragmentu nazwiska: Muhammad ibn Musa al-Chorezmi.

Algorytm gotowania jajka na miękko.



Przykład

Krok 1. Włóż jajko do gotującej się wody.

Krok 2. Zanotuj czas początkowy t_0 .

Krok 3. Odczytaj czas aktualny t .

Krok 4. Oblicz $\Delta t = t - t_0$.

Krok 5. Jeśli $\Delta t < 3 \text{ min.}$, to przejdź do kroku 3.

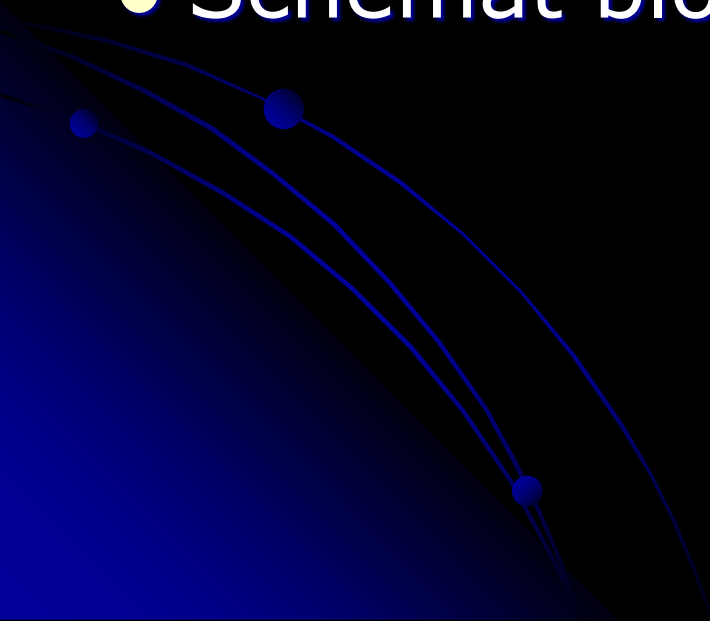
Krok 6. Wyjmij jajko z gotującej się wody.
Zakończ algorytm.

Cechy algorytmu

- ***Skończoność*** (realizowany ciąg operacji powinien mieć swój koniec).
- ***Określoność*** (zarówno operacje, jak i porządek ich wykonywania powinny być ściśle określone, nie zostawiając miejsca na dowolną interpretację użytkownika).
- ***Ogólność*** (algorytm nie ogranicza się do jednego, pojedynczego, szczegółowego przypadku, ale odnosi się do pewnej klasy zadań).
- ***Efektywność*** (algorytm powinien prowadzić do rozwiązania możliwie najprostszą drogą).

Sposoby przedstawiania algorytmów

- Opis słowny algorytmu.
- Lista kroków.
- Drzewo algorytmu.
- Schemat blokowy.



Opis słowny algorytmu



Przykład

Opracuj algorytm i jego opis słowny obliczający wartość funkcji:

Rozwiązanie

Dane: Dowolna liczba rzeczywista x .

Wynik: Wartość funkcji $f(x)$ określonej

następującym wzorem:

$$f(x) = \begin{cases} -1, & \text{dla } x < 0 \\ 0, & \text{dla } x = 0 \\ 1, & \text{dla } x > 0 \end{cases}$$

Algorytm w postaci listy kroków



Przykład

Opracuj algorytm w postaci listy kroków obliczający wartość funkcji:

Rozwiązanie

Dane: Dowolna liczba rzeczywista x .

- **Wynik:** Wartość funkcji $f(x)$

Krok 1. Wczytaj wartość danej x .

Krok 2. Jeśli $x > 0$, to $f(x) = 1$. Zakończ algorytm.

Krok 3. $\{x \leq 0\}$ Jeśli $x = 0$, to $f(x) = 0$. Zakończ algorytm.

Krok 4. $\{x < 0\}$ Mamy $f(x) = -1$. Zakończ algorytm.

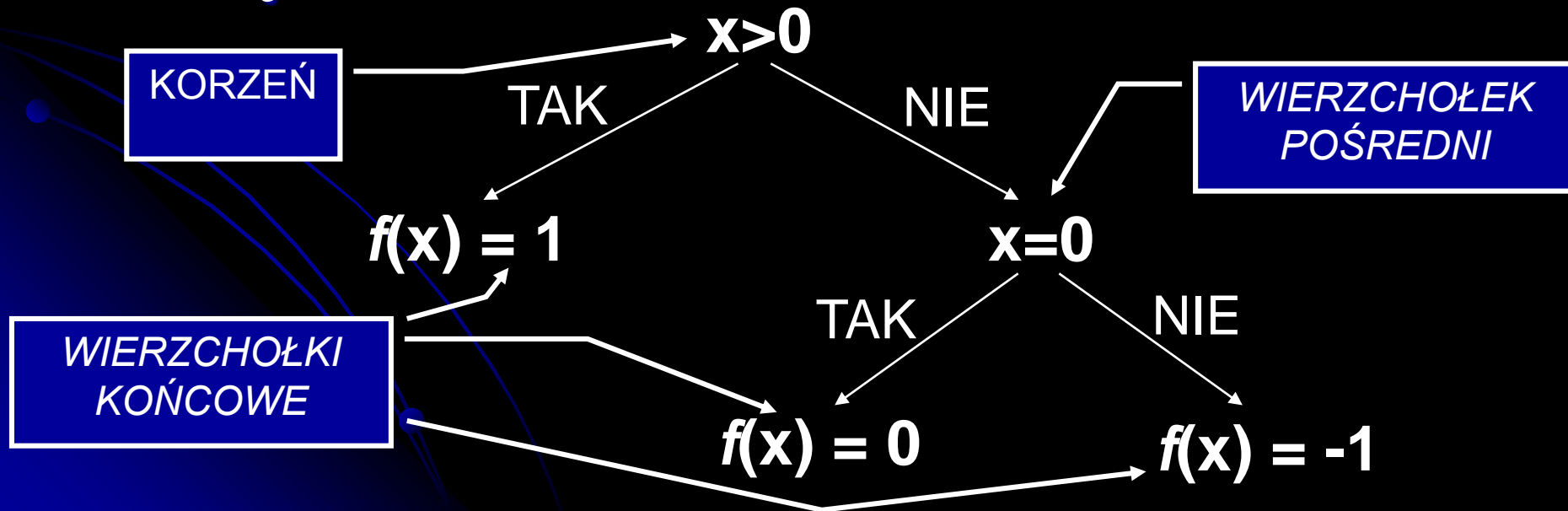
Drzewo algorytmu (drzewo obliczeń)



Przykład

Opracuj algorytm w postaci drzewa algorytmu obliczający wartość funkcji:

Rozwiązanie

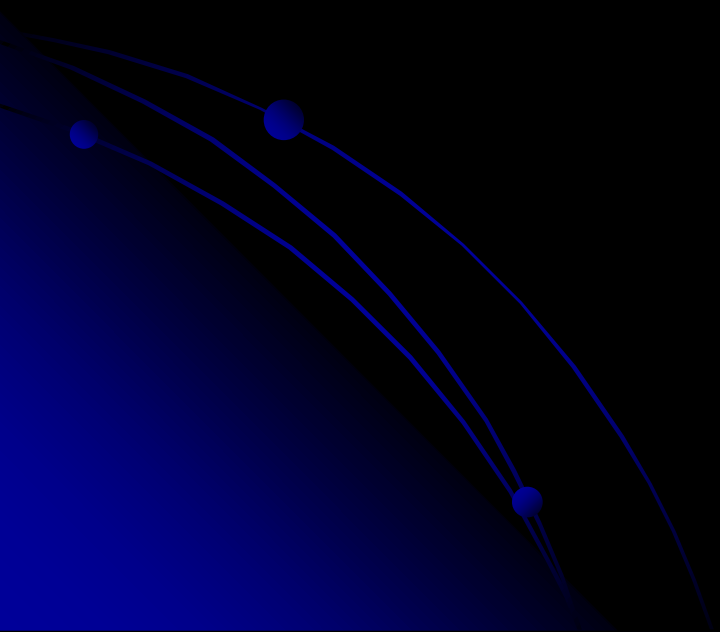


Schemat blokowy algorytmu 1/2



Przykład

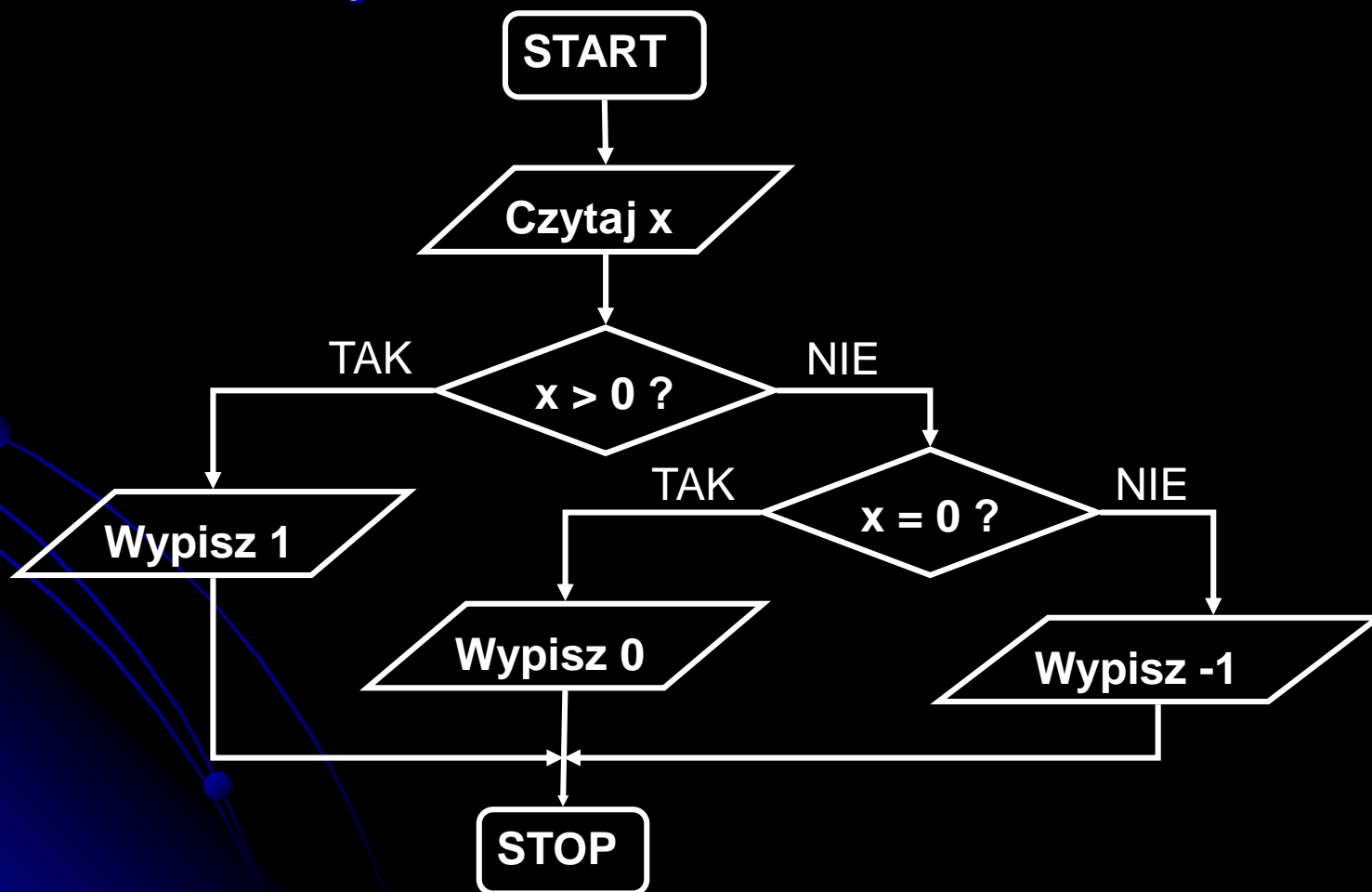
Opracuj algorytm w postaci schematu blokowego obliczający wartość funkcji:



Schemat blokowy algorytmu 2/2

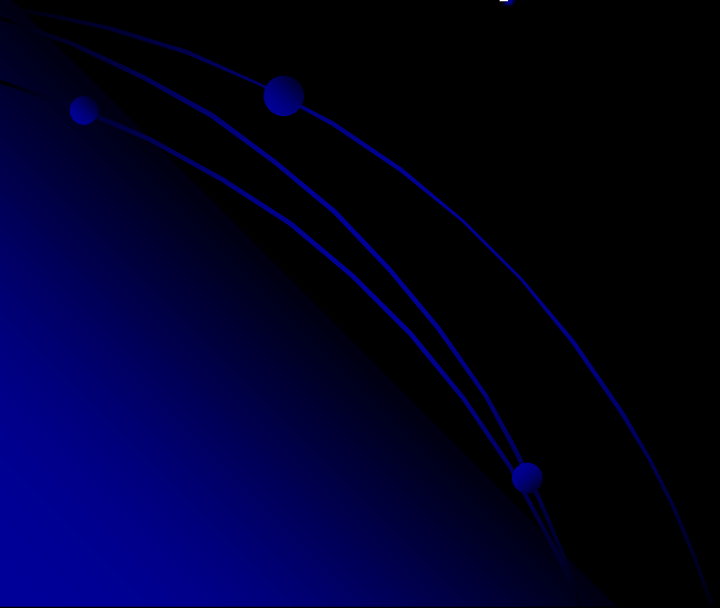


Rozwiązanie



Reguły i sposoby rysowania schematów blokowych

- Zasady budowy
- Rodzaje skrzynek
- Typowe struktury schematów blokowych

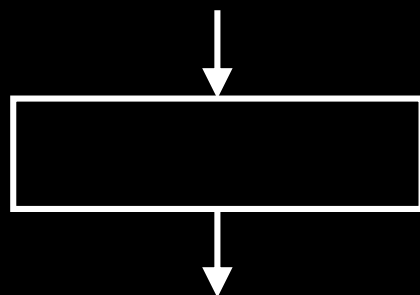


Zasady budowy

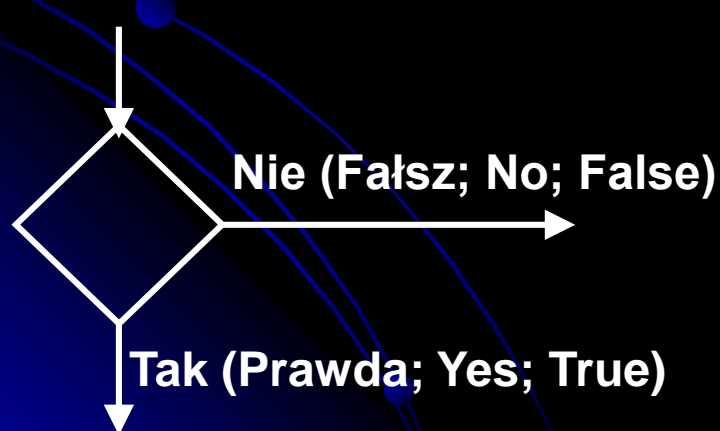
- Każda operacja, relacja lub informacja umieszczona jest w skrzynce.
- Kolejność wykonywania operacji wyznaczają połączenia między skrzynkami.
- Każde połączenie jest zaczepione początkiem do skrzynki, a końcem do innej skrzynki lub innego połączenia; żadne połączenie nie rozdziela się.
- Skrzynki przybierają kształty: prostokąta, rombu (lub sześciokąta), równoległoboku, okręgu lub owalu.
- Każdy schemat blokowy musi mieć tylko jeden element startowy oraz co najmniej jeden element końca algorytmu.

Rodzaje skrzynek 1/3

Skrzynka operacyjna

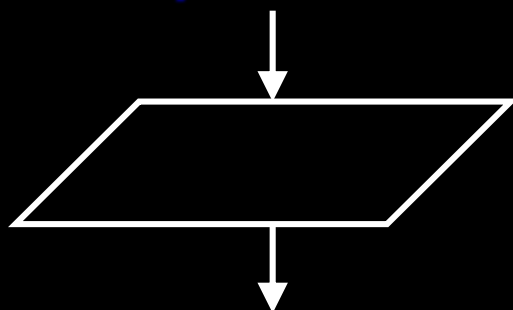


Skrzynka warunkowa (decyzyjna)

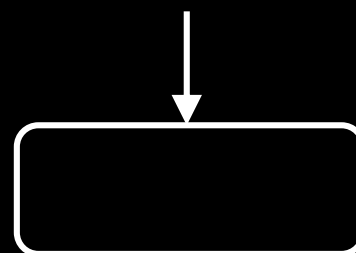
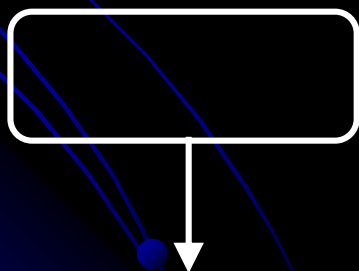


Rodzaje skrzynek 2/3

Skrzynka wprowadzania i wyprowadzania informacji (skrzynka We/Wy)

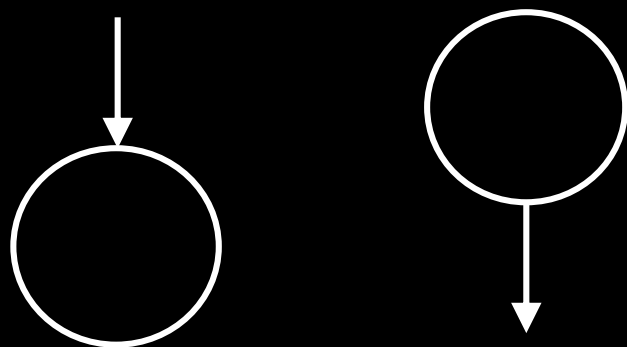


- Skrzynki graniczne START i STOP



Rodzaje skrzynek 3/3

Skrzynka łącznikowa



- Skrzynka komentarza



Wykaz słów używanych w pseudojęzyku (pseudokodzie)

- wczytaj (czytaj)
- wypisz (pisz)
- wykonaj
- jeżeli (jeśli)
- to
- w przeciwnym razie
- powtarzaj
- dopóki (aż do)

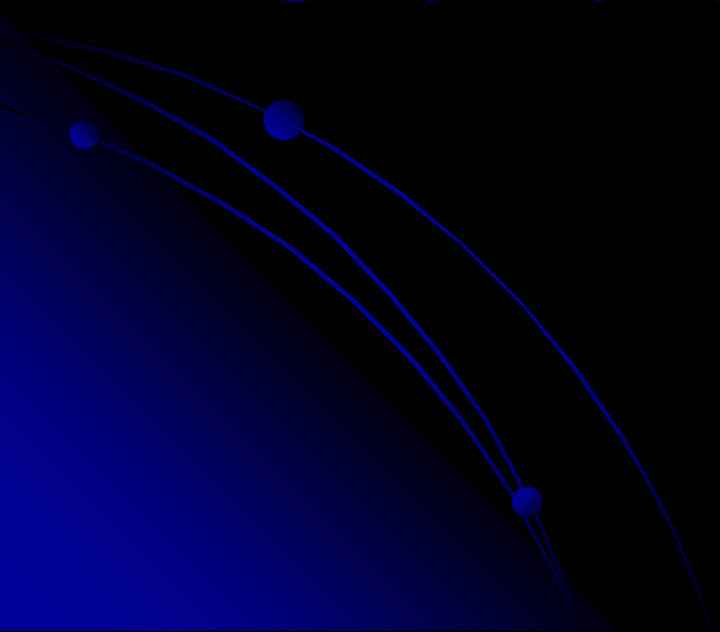
Algorytm w postaci pseudokodu

Obliczanie pewnej sumy...

1	$p1 \leftarrow 1$
2	$suma \leftarrow 0$
3	dla $k \leftarrow 1 \dots n$ wykonuj
4	$p1 \leftarrow p1 * n$
5	$p2 \leftarrow 1$
6	dla $i \leftarrow 1 \dots n$ wykonuj
7	$p2 \leftarrow p2 * k$
8	$suma \leftarrow suma + p1 + p2$

Typowe struktury algorytmów

- Algorytmy liniowe
- Algorytmy z rozgałęzieniami
- Algorytmy cykliczne (z pętlą)



Algorytmy liniowe

Algorytmem liniowym nazywamy taki algorytm, który ma postać listy kroków wykonywanych zgodnie z kolejnością ich występowania na tej liście.

- Algorytmy liniowe są zapisem obliczeń, które mają postać ciągu operacji (czynności) wykonywanych bez sprawdzania jakichkolwiek warunków.

Algorytmy liniowe



Przykład

Algorytm liniowy obliczania wartości wielomianu II stopnia postaci: $w(x) = ax^2 + bx + c = axx + bx + c$

Algorytm w pseudojęzyku:

wczytaj a, b, c, x

wykonaj:

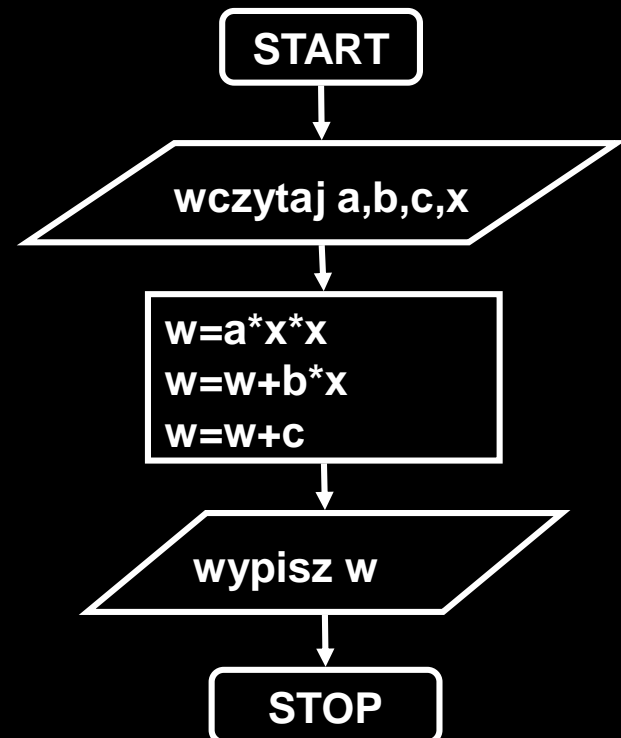
$w = a * x * x$

$w = w + b * x$

$w = w + c$

wypisz w

Schemat blokowy:



Algorytmy z rozgałęzieniami

Algorytmem z rozgałęzieniami nazywamy algorytm, w którym wykonywanie obliczeń zależy od spełnienia pewnych warunków, tzn. zawiera on w sobie wybór jednej z kilku możliwych dróg rozwiązania danego zadania. W schemacie blokowym takiego algorytmu istnieje co najmniej jedna skrzynka warunkowa.

Algorytmy z rozgałęzieniami



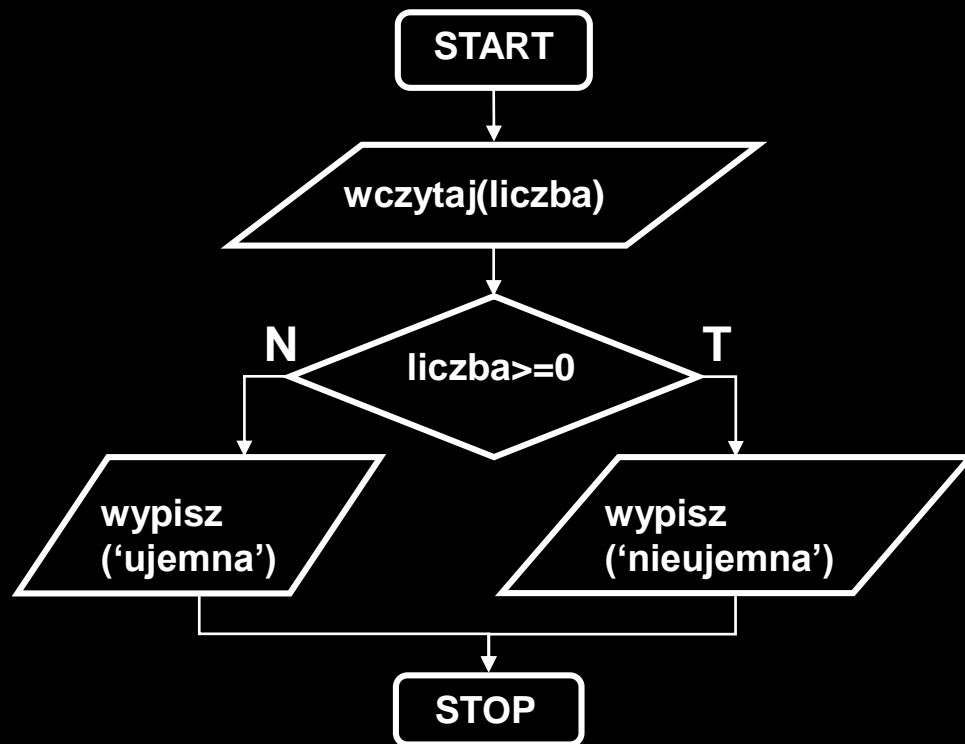
Przykład

Algorytm sprawdzania czy podana przez użytkownika liczba całkowita jest liczbą ujemną, czy nieujemną.

Algorytm w pseudokodzie:

```
wczytaj (liczba);  
jeśli (liczba  $\geq$  0)  
• wypisz ('nieujemna')  
w przeciwnym razie  
wypisz ('ujemna');
```

Schemat blokowy:



Algorytmy cykliczne (z pętlą)

Algorytmem cyklicznym nazywamy algorytm, w którym powtarzane są pewne operacje lub nawet całe fragmenty obliczeń (np. kilku kroków) jedynie dla innych zestawów danych.

Pętla obejmuje w schemacie blokowym tę część skrzynek, która ma być wielokrotnie powtarzana.

Algorytmy cykliczne (z pętlą)



Przykład

Algorytm wczytywania ciągu liczb naturalnych do momentu, gdy ich suma przekroczy wartość 100.

Rozwiązanie

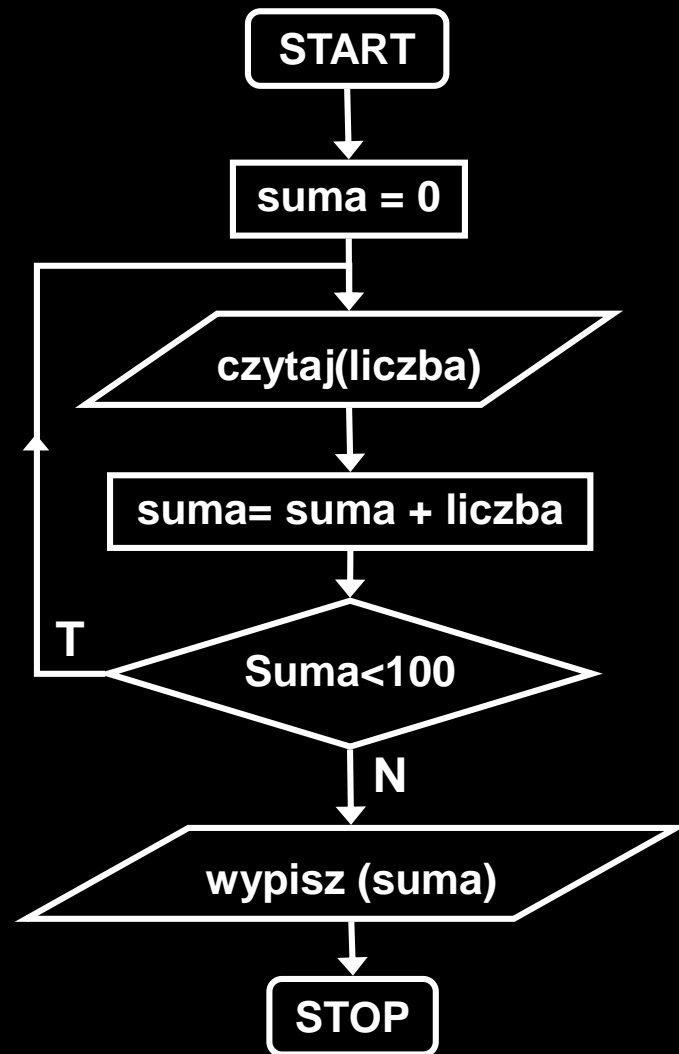
- Krok 1.** Wyzeruj sumę.
- Krok 2.** Wczytaj liczbę.
- Krok 3.** Dodaj wczytaną liczbę do sumy.
- Krok 4.** Jeżeli suma jest większa od 100 przejdź do punktu 5, w przeciwnym razie wróć do punktu 2.
- Krok 5.** Wypisz sumę wczytanych liczb.

Algorytmy cykliczne (z pętlą)

Algorytm w pseudokodzie:

```
suma = 0;  
powtarzaj  
{  
    czytaj (liczba);  
    suma = suma + liczba;  
} dopóki (suma < 100);  
wypisz (suma);
```

Schemat blokowy:



Pierwszy program w C++

```
#include<iostream>
```

```
using namespace std;
```

```
int main()  
{
```

```
//tutaj umieszczamy deklaracje zmiennych  
//operacje itp..
```

```
return 0;  
}
```



Typy danych w języku C++

Nazwa	Znak	Rozmiar w bajtach	Minimalna wartość	Maksymalna wartość
short	signed unsigned	2	-32768 0	32767 65535
int long int	signed unsigned	4	-2 147 483 648 0	2 147 483 647 4 294 967 295
long long	signed unsigned	8	-2^{63} 0	$2^{63}-1$...

Przykłady użycia typów danych w C++

```
int a;
```

Deklaracja zmiennej całkowitej o nazwie a;

```
int liczba1=0;
```

Deklaracja i inicjalizacja wartością 0 zmiennej całkowitej o nazwie liczba1;

```
float b = 3.5;
```

Deklaracja i inicjalizacja wartością 0 zmiennej całkowitej o nazwie liczba1;

```
char z;
```

Deklaracja zmiennej znakowej o nazwie z;

```
char znak='P';
```

Deklaracja i inicjalizacja znakiem 'P' zmiennej znakowej o nazwie znak;

Każda prosta instrukcja kończy się średnikiem ;

Deklaracje i inkrementacje

// ustawienie wartości początkowej zmiennej

int a = 7;

a:

7

a = 9; *// zmiana wartości zmiennej a n 9*

9

a = a+a; *// podwojenie wartości a*

18

a += 2; *// dodanie do a wartości 2, a=a+2*

20

a++; *// inkrementacja, czyli zwiększenie o 1*

21

Operatory relacji w języku C++

Podstawowe operatory wykorzystywane do porównywania wartości zmiennych

Zapis matematyczny	Zapis w języku C++	Opis
=	==	równy np., <code>if(a==b)</code> [<code>(a=b)</code> oznacza wstaw <code>b</code> w miejsce <code>a</code>]
<	<	mniejszy niż
≤	<=	mniejszy lub równy
>	>	wiekszy
≥	>=	wiekszy lub równy
≠	!=	różny

Operatory logiczne

- Tabele prawdy dla operatora "&&" (i) oraz „||” (lub):

Warunek1, np. X>0	Warunek2, np. X<10	Warunek1 && Warunek2	Warunek1 Warunek2
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Elementy języka programowania Java

Deklaracja zmiennych

<typ wartości> <lista nazw>;

```
int i, j;
```

Wybrane dostępne typy:

long ,int, short byte,	(całkowity)
float, double,	(rzeczywisty)
char	(znakowy)

Referencja do tablicy:

<typ wartości> <nazwa typu> []; (tablica jednowymiarowa)

```
int tab[];
```


Komentarze w kodzie programu

Komentarze w kodzie programu są ciągami znaków pomijanymi przez kompilator.

Komentarz blokowy:

/*

to jest komentarz blokowy, który może
zajmować wiele linii

*/

Komentarz liniowy:

// to jest komentarz liniowy

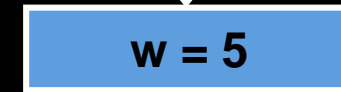
Elementy języka programowania C++ a schematy blokowe

Instrukcja przypisania:

int w;

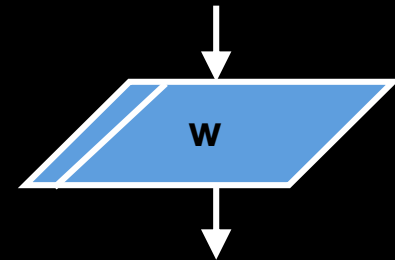
w = 5;

gdzie: *x* - nazwa zmiennej,
w - wyrażenie.



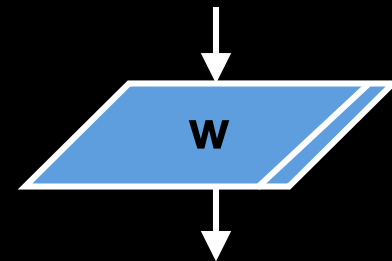
Instrukcja wczytywania

cin >> w;



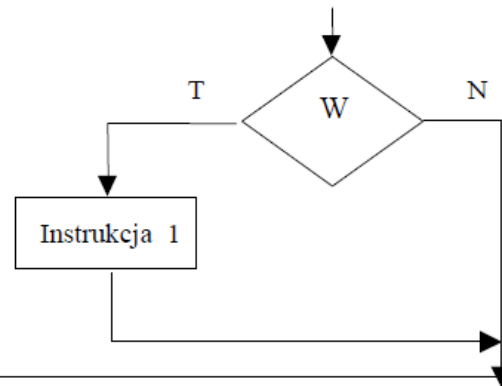
Instrukcja wyjścia:

cout << w;

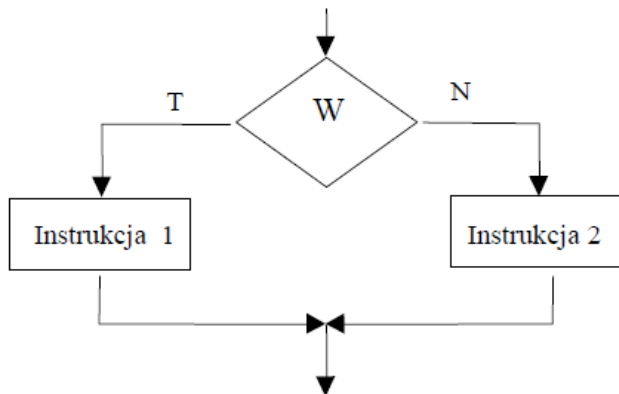


Instrukcja warunkowa if oraz if..else

```
if(wyrażenie warunkowe W)
{
    //instrukcje 1
}
```



```
if(wyrażenie warunkowe W)
{
    //instrukcje 1
}
else
{
    //instrukcje 2
}
```



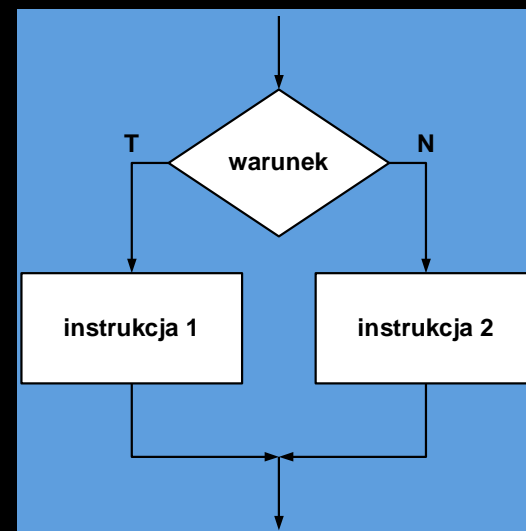
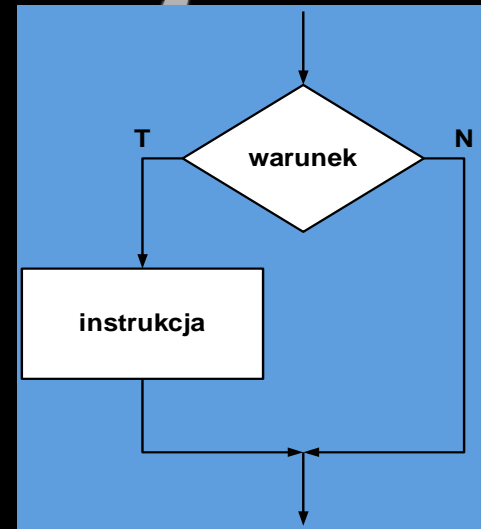
Elementy języka programowania

C++- instrukcje

Instrukcje selekcji (warunkowe):

```
if (warunek)  
{  
    instrukcja;  
}
```

```
if ( warunek)  
{  
    instrukcja1;  
} else  
{  
    instrukcja2;  
}
```



Elementy języka programowania

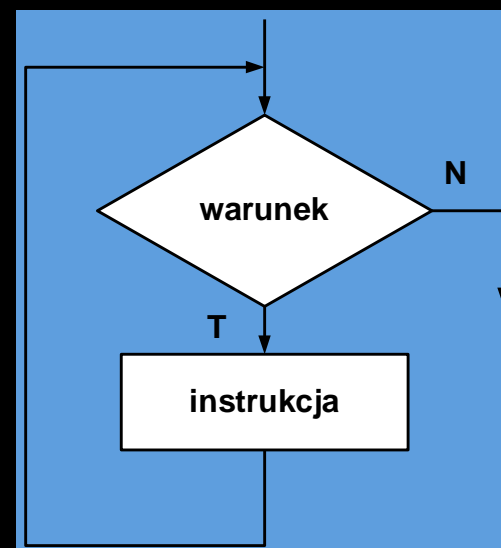
C++ - pętle

Instrukcje pętli (iteracji)

```
while (warunek)  
{  
    instrukcja;  
}
```

W wersji polskiej:

**dopóki warunek wykonuj
instrukcja**



Elementy języka programowania

C++ - pętle

Instrukcje pętli (iteracji)

do {

instrukcja;

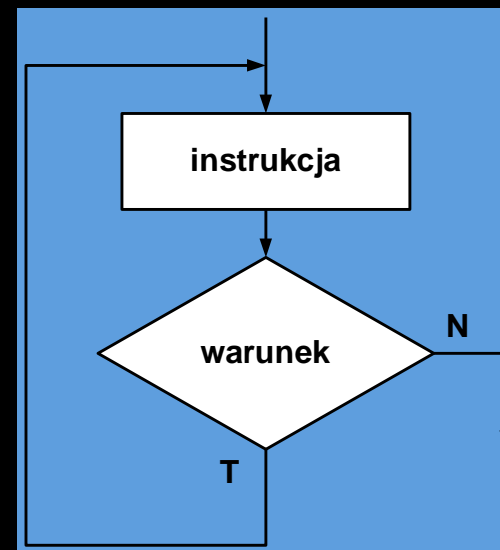
} while (warunek);

W wersji polskiej:

powtarzaj

instrukcja;

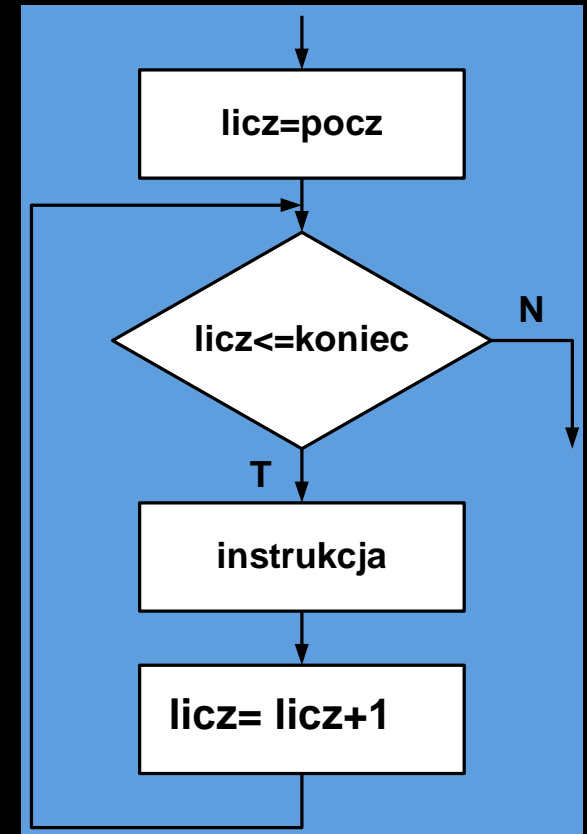
dopóki warunek



Elementy języka programowania Java - instrukcje

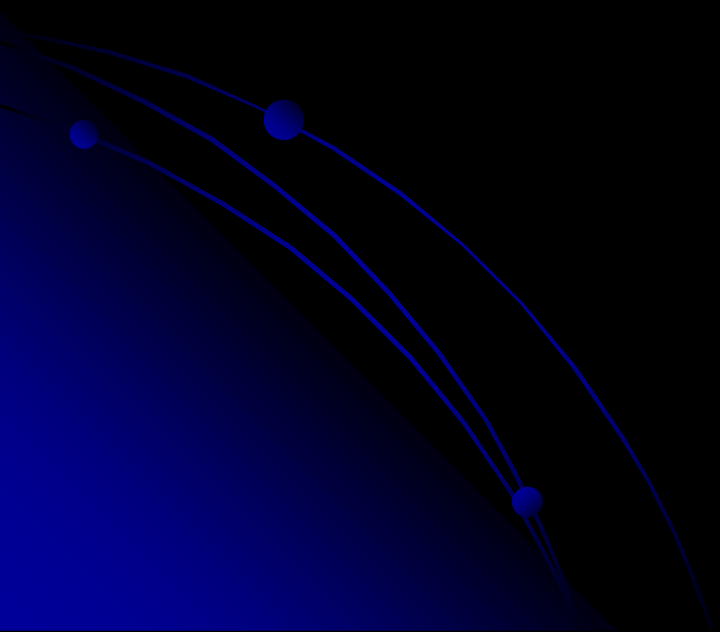
Instrukcja **for** jako szczególny przypadek instrukcji pętli

```
for (licz=pocz; licz<=koniec ; licz=licz+1)  
{  
    instrukcja;  
}
```



Typowe struktury programów w języku Java

- Programy liniowe
- Programy z rozgałęzieniami
- Programy cykliczne





Przykład: Program liniowy.

Obliczanie wartości wielomianu II stopnia: $w(x) = ax^2 + bx + c = axx + bx + c$.

Zapis algorytmu w języku C++:

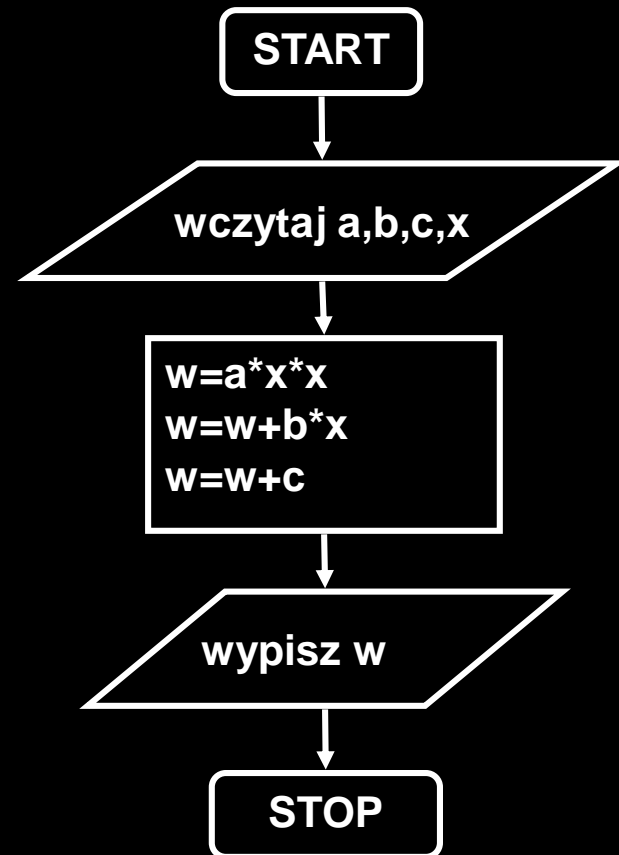
```
#include <iostream>

using namespace std;

int main()
{
    int a, b, c, x;
    cin >> a >> b >> c >> x;
    int w = a*x*x;
    w = w + b*x;
    w = w + c;
    cout << w << endl;

    return 0;
}
```

Schemat blokowy:





Przykład: Program z rozgałęzieniami.

Algorytm sprawdzania czy podana przez użytkownika liczba całkowita jest liczbą ujemną, czy nieujemną.

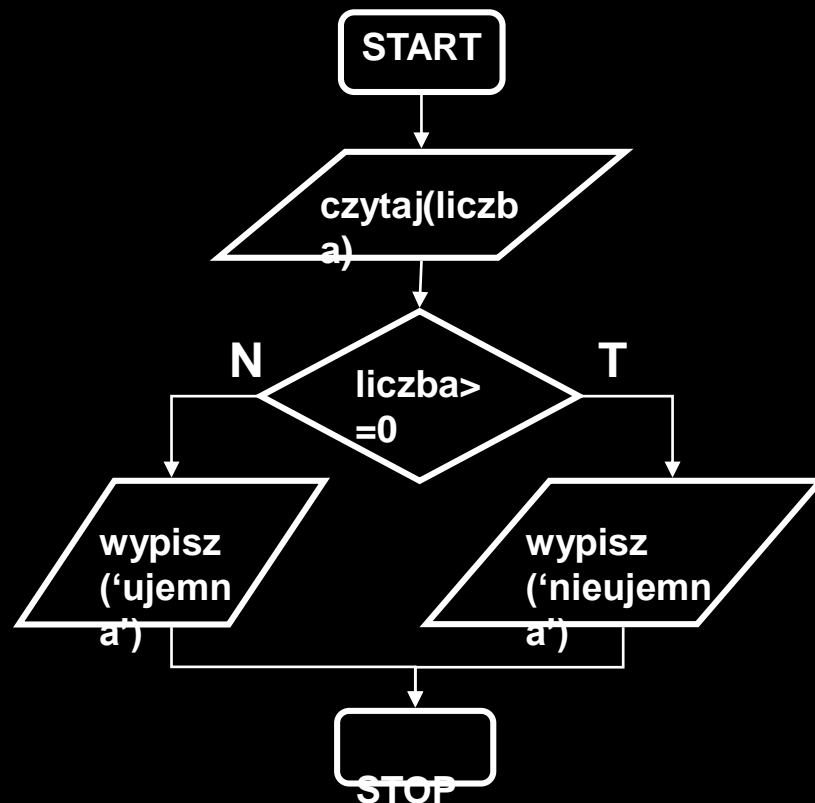
Zapis algorytmu w języku C++:

```
#include <iostream>
using namespace std;

int main()
{
    int liczba;
    Cin>>liczba;

    if (liczba>=0)
        cout<<"nieujemna";
    else
        cout<<"ujemna";
}
```

Schemat blokowy:





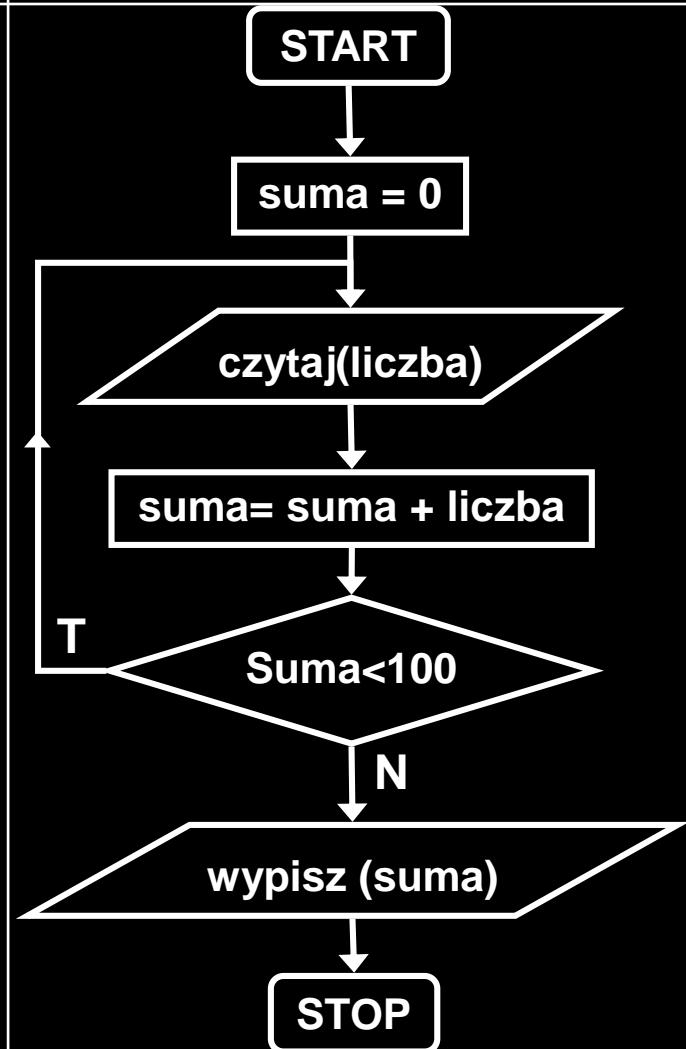
Przykład: Program cykliczny (nieznana liczba powtórzeń).
Wczytywania ciągu liczb naturalnych do momentu, gdy ich suma przekroczy wartość 100.

Zapis algorytmu w języku C++:

```
#include <iostream>
using namespace std;
int main()
{
    int liczba,suma;
    suma=0;
    do
    {
        cin>>liczba;
        suma = suma + liczba;
    } while (suma<100);

    cout<<"suma=" +suma;
    return 0;
}
```

Schemat blokowy:



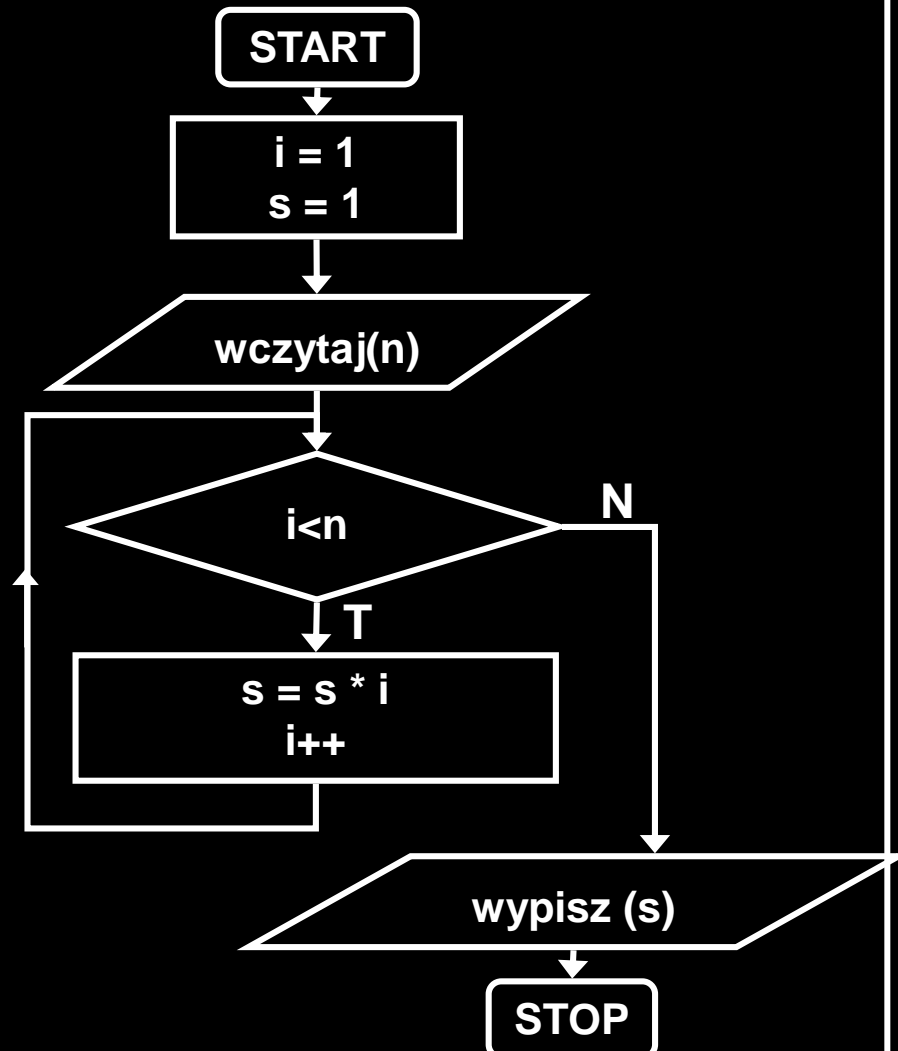


Przykład: Program cykliczny (znana liczba powtórzeń).
Obliczanie silni s liczby całkowitej nieujemnej n .

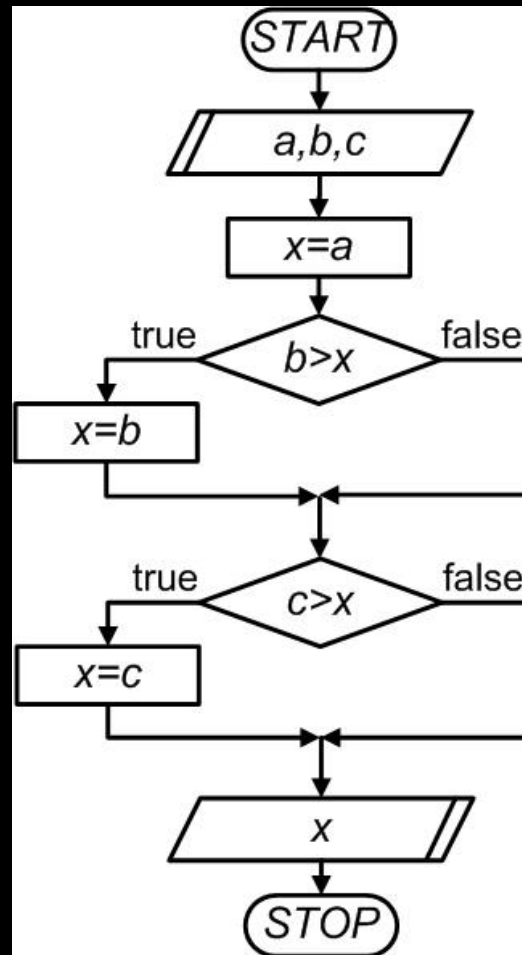
Zapis algorytmu w języku C++:

```
#include <iostream>
using namespace std;
int main()
{
    int n,i,s;
    i=1;s=1;
    cin>>n;
    while (i<n)
    {
        s=s*i;
        i++;
    }
    cout<<"s="<<s;
    return 0;
}
```

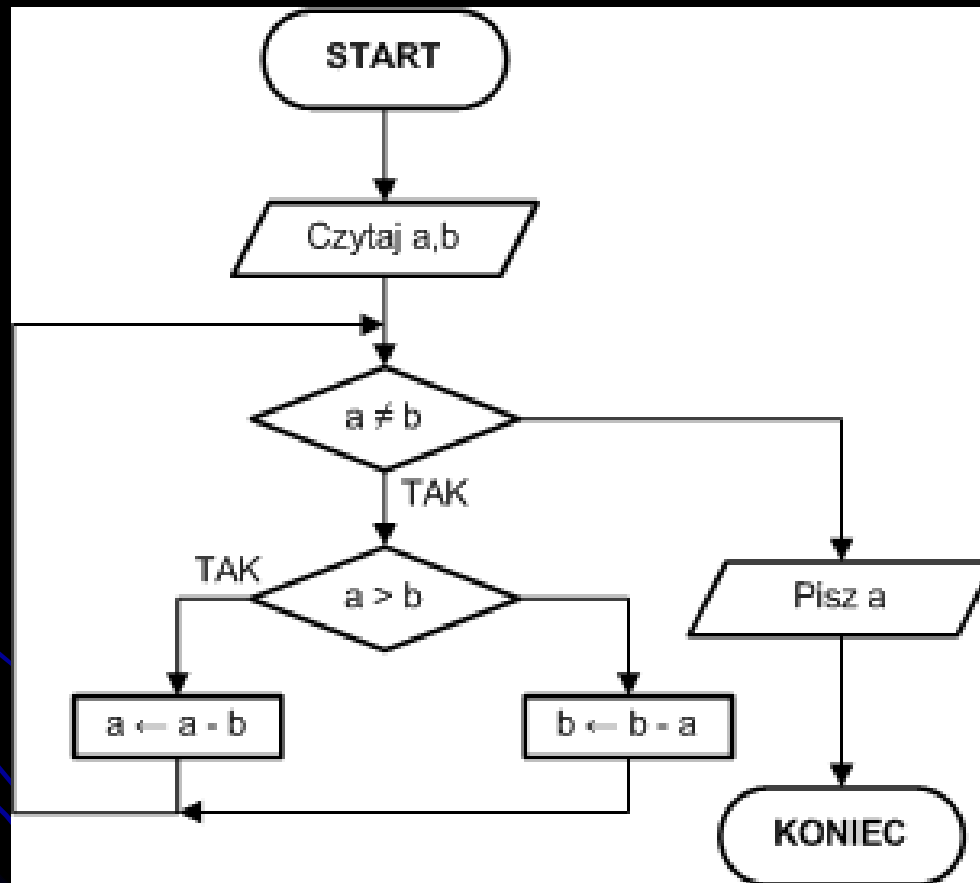
Schemat blokowy:



Zapisz schemat blokowy w języku C++:



Jakie zadanie realizuje ten algorytm?



Jakie zadanie realizuje ten algorytm?

