

INFORMATYKA

C

WYDZIAŁ ELEKTROTECHNIKI I
INFORMATYKI
ELEKTROTECHNIKA
DR INŻ. MICHAŁ ŁANCZONT

II

Etapy tworzenia programu



Algorytm można zdefiniować na wiele sposobów, ujmując najprościej to skończony, uporządkowany ciąg określonych działań lub czynności koniecznych do wykonania określonego zadania.

- **opisowej** (słownie tzw. Pseudokod)
- **graficznej**:
 - schematy blokowe (ang. flowcharts),
 - schematy Nassi-Schneidermana (schematy N/S lub zwarte).

Przygotowanie optymalnego algorytmu pozwala na sprawną implementację zaproponowanego rozwiązania w wybranym języku programowania.

1. poprawny, tzn. dla każdego poprawnego zestawu danych, po wykonaniu skończonej liczby czynności, prowadzi do poprawnych wyników.
2. jednoznaczny- w każdym wypadku jego zastosowania, dla tych samych danych uzyskamy ten sam wynik
3. szczegółowy, aby wykonawca algorytmu rozumiał opisane czynności i potrafił je wykonać
4. uniwersalny, aby służył do rozwiązywania pewnej grupy zadań, a nie tylko jednego konkretnego przypadku zadania

Etapy konstruowania algorytmu

1. sformułowanie zadania – ustalamy jaki problem ma rozwiązywać algorytm
2. określenie typu danych wejściowych – w typie określamy, czy dane są liczbami rzeczywistymi, całkowitymi, czy znakami, czy też innego typu.
3. określenie wyniku oraz sposobu jego prezentacji
4. ustalenie metody wykonania zadania (może być kilka metod na rozwiązanie – wybieramy optymalną)

Etapy konstruowania algorytmu

5. zapisanie algorytmu za pomocą wybranej metody
6. analiza poprawności rozwiązania
7. testowanie rozwiązania dla różnych danych (algorytm musi być uniwersalny, aby służyć do rozwiązywania zadań dla różnych danych wejściowych)
8. ocena skuteczności algorytmu (praktyczna ocena algorytmu : np. szybkości, skomplikowania).

Schematy blokowe

- Schemat blokowy jest narzędziem prezentującym w graficznej postaci kolejność działań (czynności) w tworzonym algorytmie.
- Składa się z figur geometrycznych (np. prostokąt, romb, owal) prezentujących różne działania oraz strzałek wskazujących powiązania między elementami schematu oraz kierunek przepływu informacji.

Schematy blokowe

- Elementy składowe, ich forma graficzna i znaczenie określone są w odpowiednim standardzie opisanym normą
- ISO 5807:1985 Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts

- Zapisując poszczególne zadania realizowane w ramach poszczególnych elementów algorytmu należy zapisywać łącznie
- Ograniczone miejsce w elementach graficznych schematu blokowego i NS uniemożliwia stosowanie opisu słownego

Operatory arytmetyczne

Symbol	Opis
+	Dodawanie
-	Odejmowanie
*	Mnożenie
/	Dzielenie
^	Potęgowanie
sqrt()	Pierwiastkowanie
DIV	Dzielenie całkowite
MOD	Reszta z dzielenia

Operatory relacji

Symbol	Opis
<code>=</code> lub <code>==</code>	Równy
<code><></code> lub <code>!=</code>	Różny
<code><</code>	Mniejszy
<code>></code>	Większy
<code>>=</code>	Większy lub równy
<code><=</code>	Mniejszy lub równy

Pozostałe operatory

Symbol	Opis
= lub :=	Przypisanie
AND lub &	Logiczne „i”
OR lub 	Logiczne „lub”
NOT lub ~, !	Logiczna negacja „nie”

X	Y	X&Y	X Y	X	!X
TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE		
FALSE	FALSE	FALSE	FALSE		

Schemat blokowy

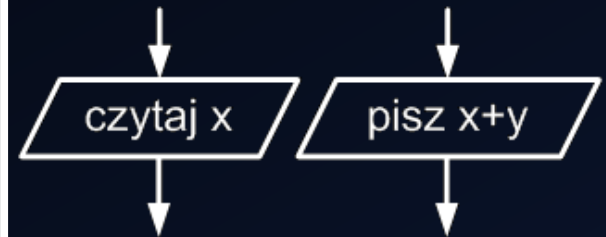
Graniczny

Przedstawia początek, koniec, przerwanie lub wstrzymanie wykonywania działania



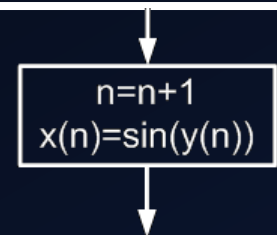
Wejścia-Wyjścia

Wprowadzania danych do programu lub przedstawianie wyników działania programu



Operacyjny

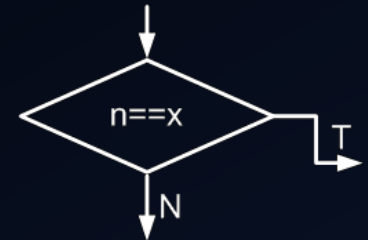
Wykonywania operacji w efekcie których zmieniają się wartości, postać lub miejsce zapisu danych



Schemat blokowy

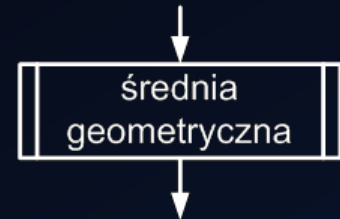
Decyzyjny, warunkowy

Pokazuje wybór jednej z dwóch dróg wykonania programu na podstawie warunku wstępnego



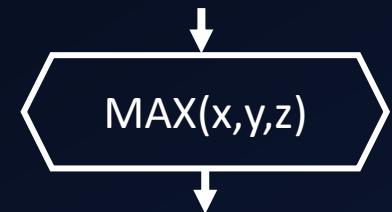
Blok fragmentu

Pokazuje wykonanie podprogramu zdefiniowanego odrębnie



Blok wywołania podprogramu

Oznacza zmianę wykonywanej czynności na skutek wywołania podprogramu



Schemat blokowy

Komentarza

Komentarz objaśniający fragment schematu



Łącznik wewnętrz-stronicowy

Umożliwia łączenie odrębnych części schematu w ramach tej samej strony, powiązania mają to samo oznaczenie



Łącznik między-stronicowy

Umożliwia łączenie odrębnych części schematu zapisanych na różnych stronach, powiązania mają w nazwie symbol oznaczenie i numer strony



Schemat blokowy

1. Schemat blokowy składa się z bloków połączonych zorientowanymi liniami;
2. Bloki obrazują ciąg operacji; powinien być start i koniec;
3. Zawsze wykonywane są albo wszystkie instrukcje w bloku albo żadna;
4. Dalsze operacje nie zależą od poprzednich wariantów, chyba że zależności te zostały przekazane za pomocą danych;

Schemat blokowy

5. Kolejność wykonania operacji jest ściśle określona przez zorientowane linie łączące poszczególne bloki;
6. Do każdego bloku może prowadzić co najwyżej jedna linia;
7. **Linie mogą się łączyć** ale nie mogą się rozdzielać (bez bloku decyzyjnego).
8. Warunki definiowane za pomocą operatorów logicznych `==`, `<`, `>`, `<=`, `>=`, `OR`, `AND`

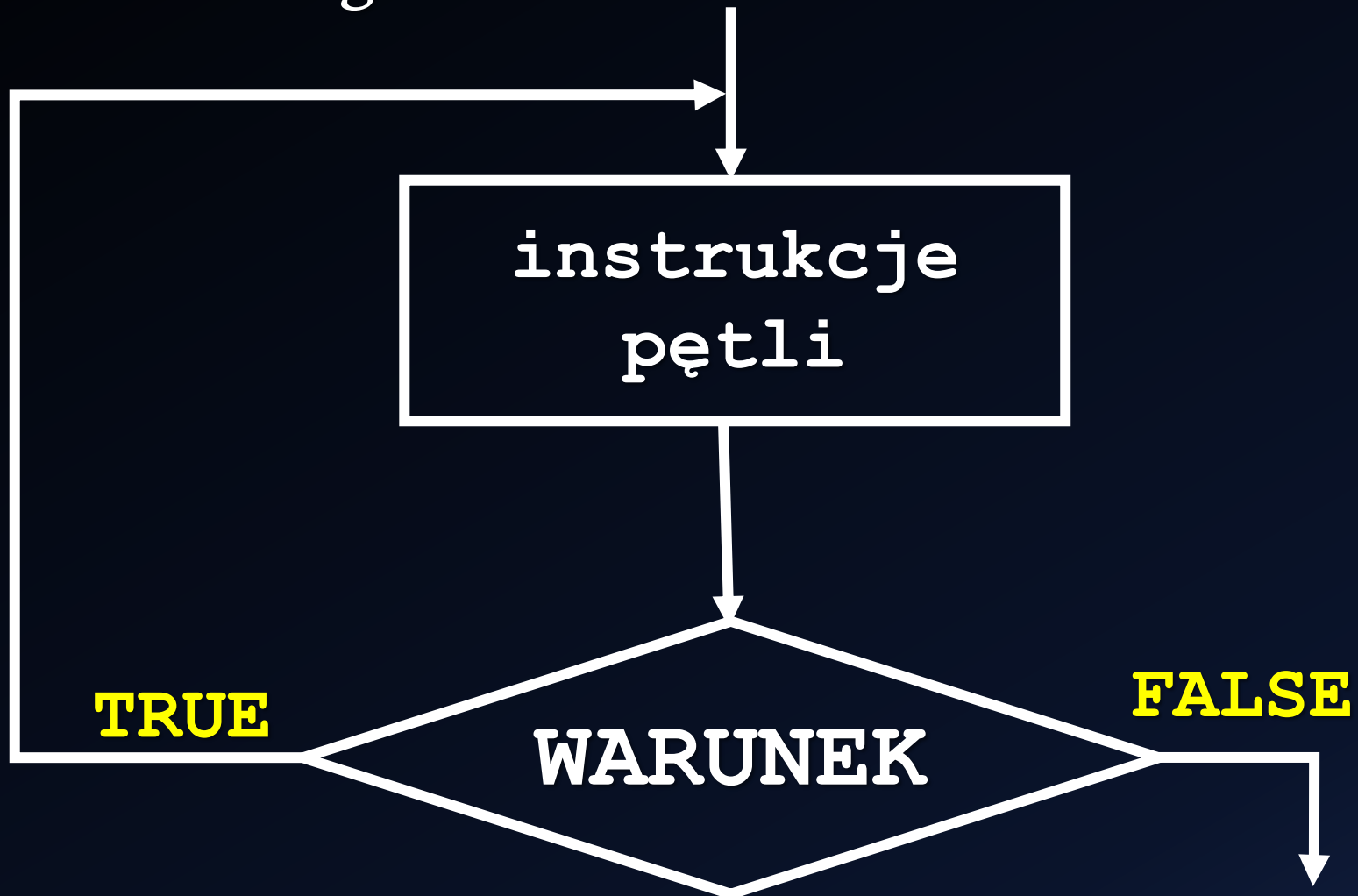
Schemat blokowy

Realizacja pętli **WHILE**, powtarzanego warunku



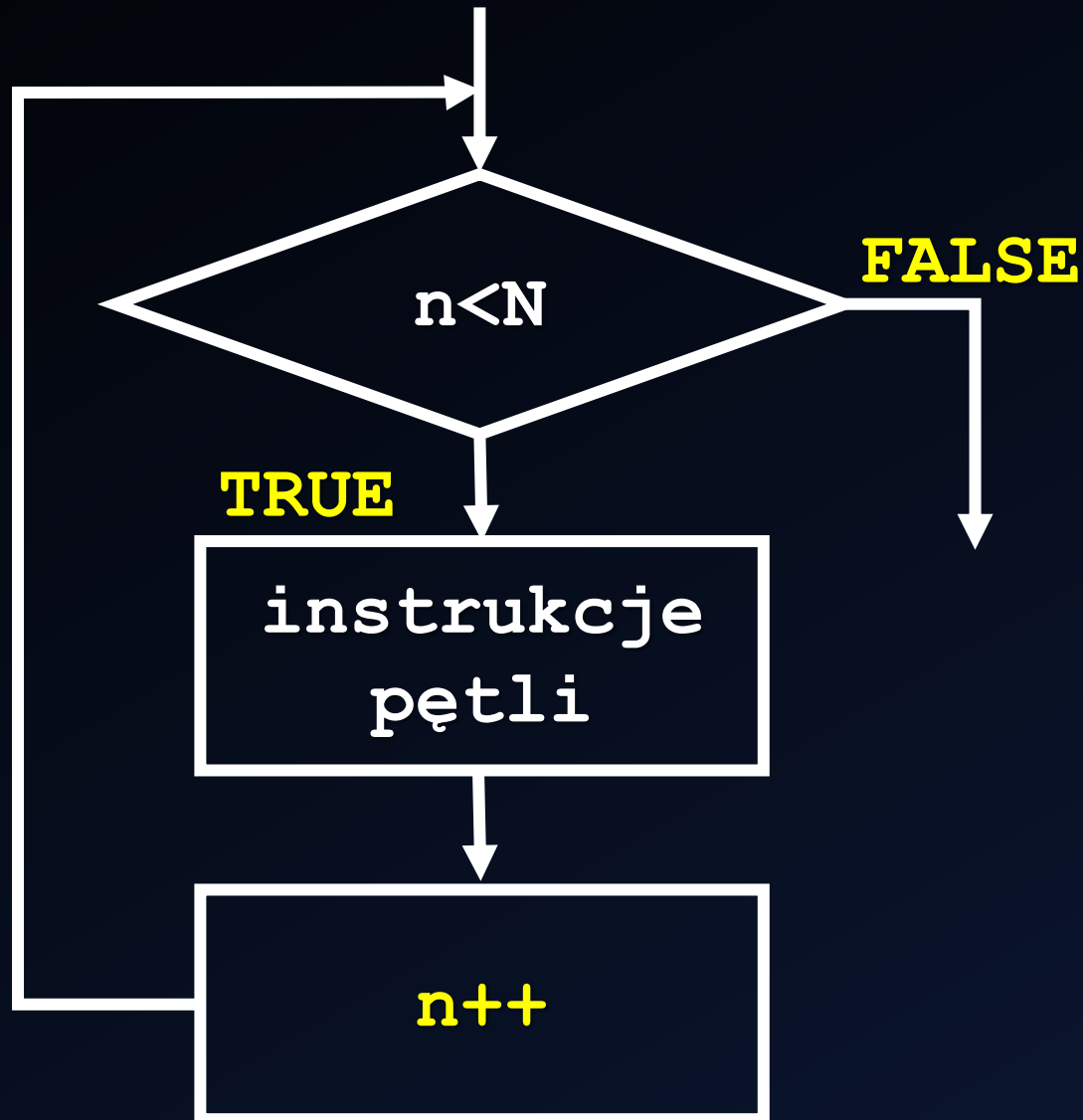
Schemat blokowy

Realizacja pętli **DO WHILE**, powtórzenia warunkowego



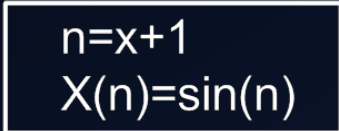
Schemat blokowy

Układ pętli **FOR** powtarzającej **N** razy



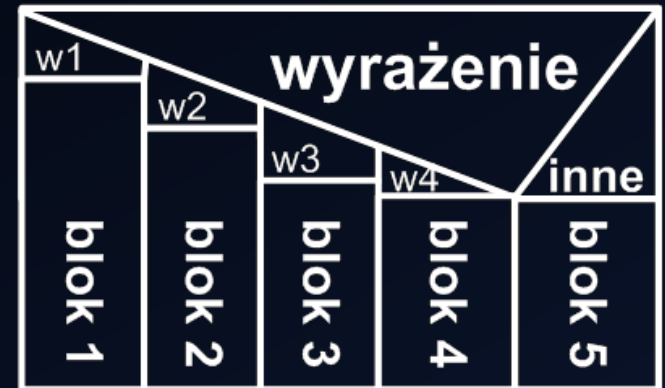
Schemat Nassi-Schneidermana

- **Schemat N/S** jest graficznym ukazaniem kolejności działań w opracowywanym algorytmie.
- Tworzony jest schemat zwarty.
- Nie występują jawne połączenia elementów (brak strzałek).
- Domyślny kierunek powiązania elementu biegnie z góry na dół.

Blok wejścia	Wprowadzanie danych, z klawiatury, pliku	
Blok wyjścia	Wyprowadzanie danych, ekran, druk, plik	
Blok operacyjny	Wykonywanie operacji	
Blok decyzyjny	Realizacja wyboru na podstawie określonego warunku	

Wybór jeden z wielu

Realizuje wybór jednego z wielu wariantów wskazując na odpowiadające im bloki wykonawcze



Wywołanie procedury lub funkcji

Wywołuje odrębnie opisaną procedurę lub funkcję



Blok iteracyjny	Cyklicznie wykonywany blok poleceń zgodnie z określonym ciągiem $i=1,2,\dots,n$	$i=1..n$ Blok wykonywany w pętli
Blok powtórzeń WHILE	Cyklicznie wykonuje zdefiniowany blok poleceń po spełnieniu warunku	warunek Blok poleceń
Blok powtórzeń DO WHILE	Cyklicznie wykonuje zdefiniowany blok poleceń sprawdzając warunek powtórzenia po wykonaniu bloku	Blok poleceń warunek

Narzędzia wspomagające

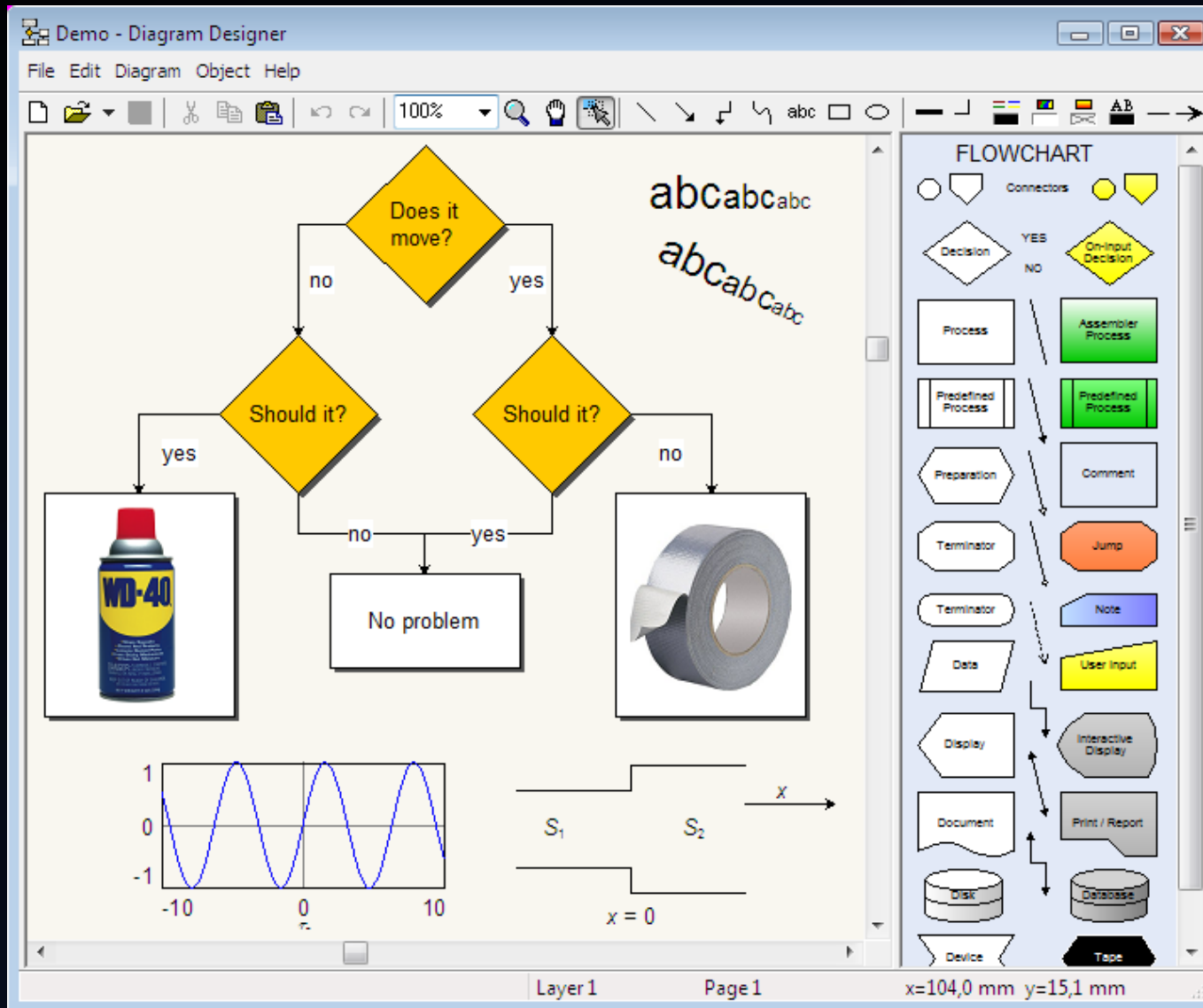
Programy komputerowe wspomagające tworzenie schematów blogowych

- Program graficzny dla grafiki rastrowej (paint)
- Programy graficzne wektorowe
 - Corel Draw
 - Microsoft Office (WORD, POWER POINT, VISIO)
 - Microsoft WhiteBoard
 - OpenOffice

Narzędzie wspomagające

- Programy komercyjne
 - Wersje trial
- Programy darmowe
 - Diagram designer
 - Lucidchart - online
 - draw.io –online
 - NS Builder
 - Code::Blocks

Diagram Designer



LucidChart

The screenshot displays the LucidChart web application interface. At the top, the Lucidchart logo is on the left, and navigation links for "DOCUMENTS", "INTEGRATIONS", and "HELP" are on the right. A "Send Invite" button is also present. Below the navigation bar, a row of template icons includes "Blank", "Flowchart", "Wireframe", "Site Map", "Android", and "iOS", followed by a "More Templates" button. The main workspace shows a grid of document thumbnails, including "Product Education", "Advanced Webinar Docs", "Demo Diagrams", "Create: Task #1", "Blank Diagram", "Umbrella Product Launch Process", "Blank Diagram", "Bubble Mind Map", "Animal/Human Venn Diagram", "Network Diagram", "Office Floorplan", "AWS Infrastructure", "Ecommerce Org Chart", "Amplifier Circuit Diagram", and several others. A sidebar on the left contains "My Documents" (with sub-items like "Advanced Webinar Docs", "Demo Diagrams", "Product Education", "Lucidchart 201", "Sample Diagrams", "Team Avocet", "Webinars"), "Recent Documents", "Shared with Me (102)", "Trash", and a "Search Results" section. At the bottom, a "Related" section lists "Contact Sales", "Free Amazon Gift Card", "Try Lucidpress", and "Integrate with Quip". A "sample diagram (1).xml" file is visible in the bottom left corner, and a "Show all" button is in the bottom right.

Step 2 of 3

Create your account



Use 6 or more characters

Create account

or continue with



Google

By registering, you agree to our [Terms of Service](#) and [Privacy Policy](#).

Sign up free

"I don't think enough people know how helpful this technology can be, but I do—I drew some of the original architecture and product sketches for Okta using Lucidchart."



Todd McKinnon
CEO and Co-Founder 

Lucid

My Documents Search documents 🔍

+ New

Home

Documents

My Documents

Shared with Me

Recent Documents

Starred Items

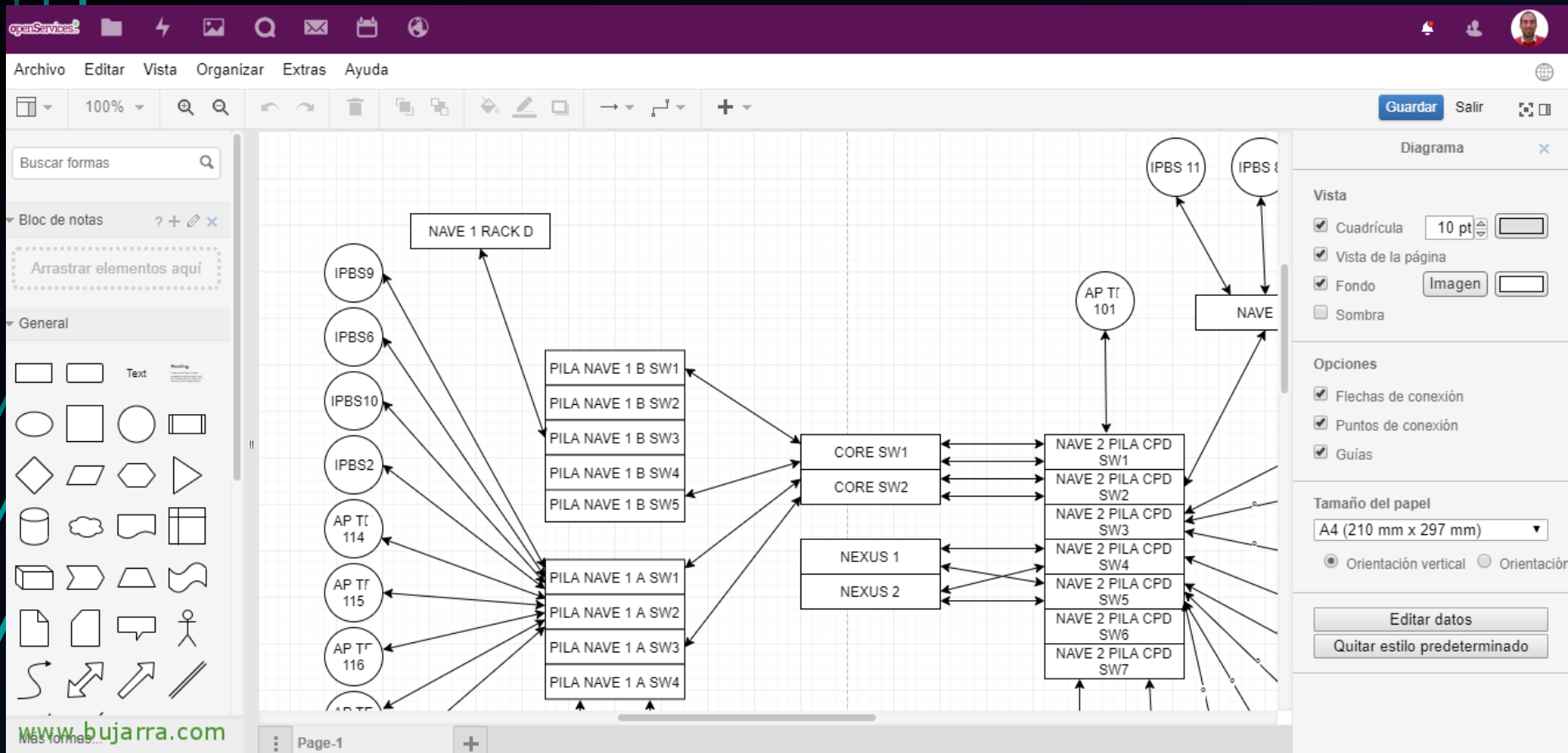
Templates

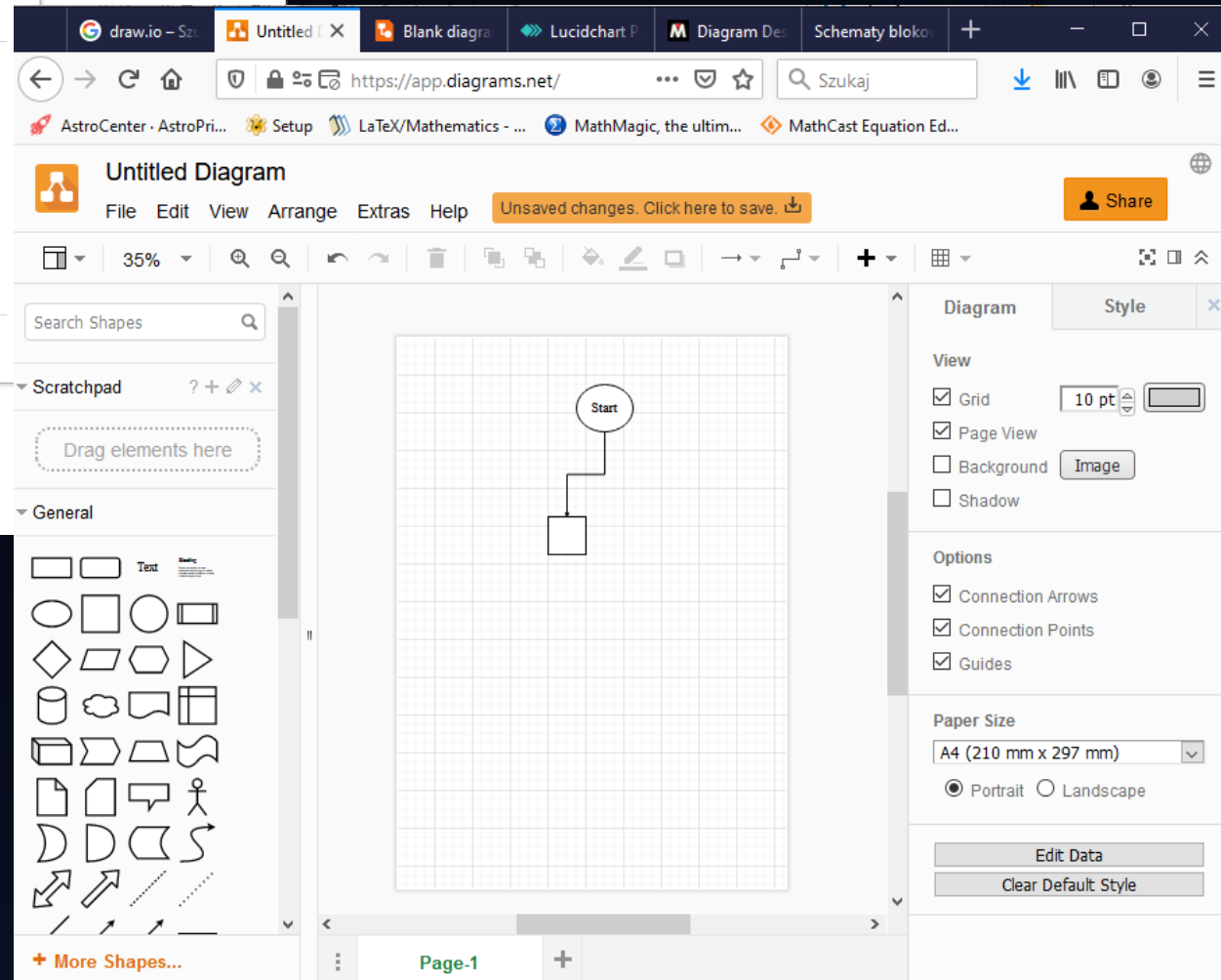
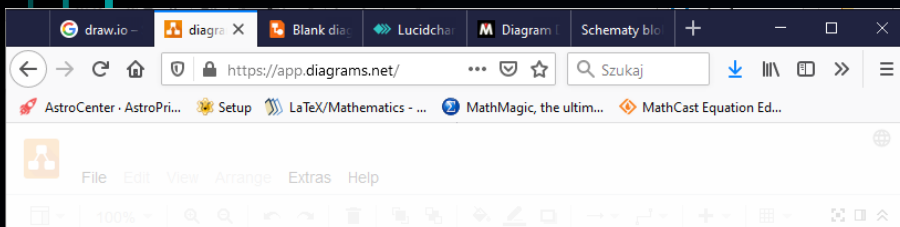
Integrations

Blank diagram

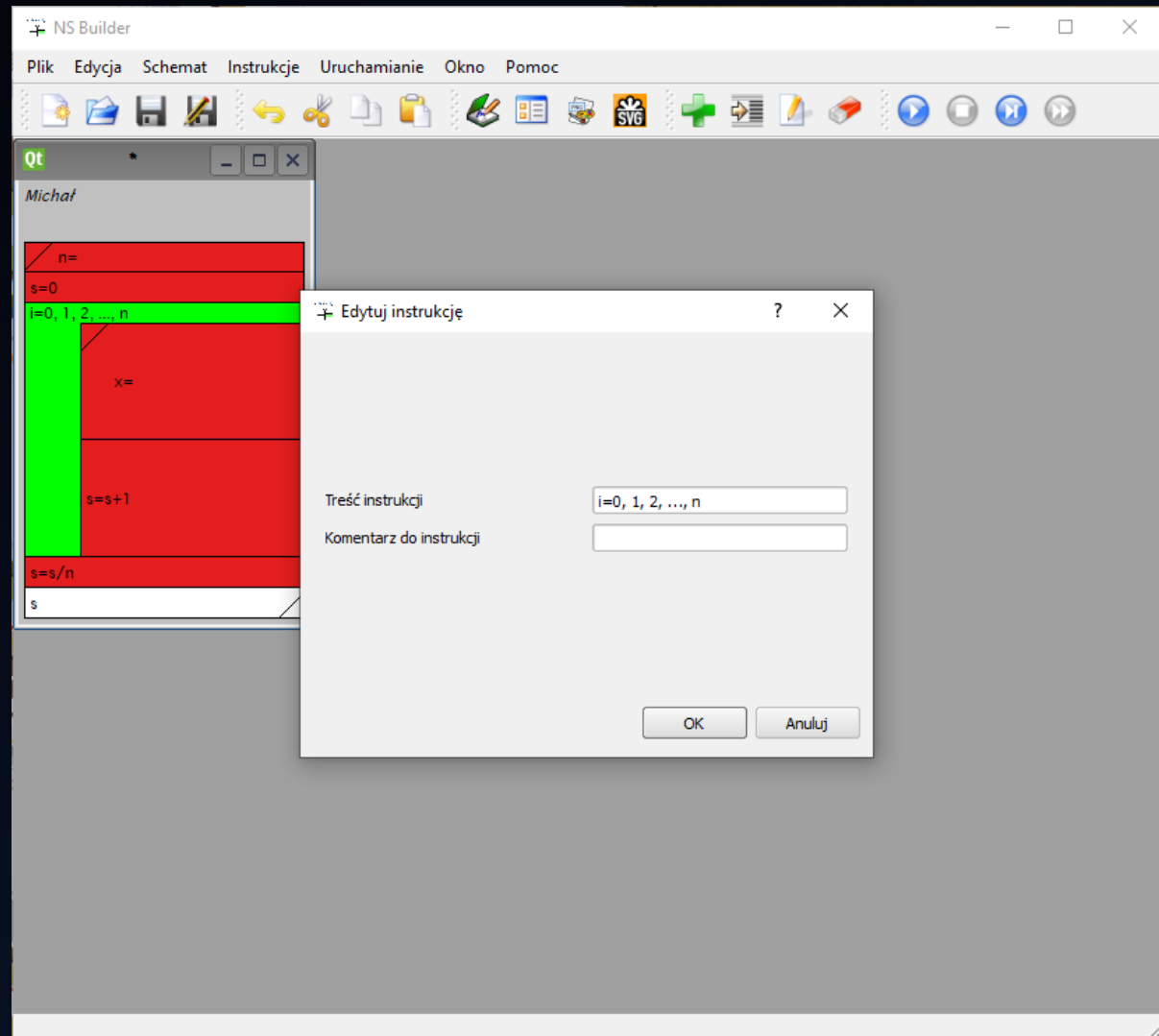
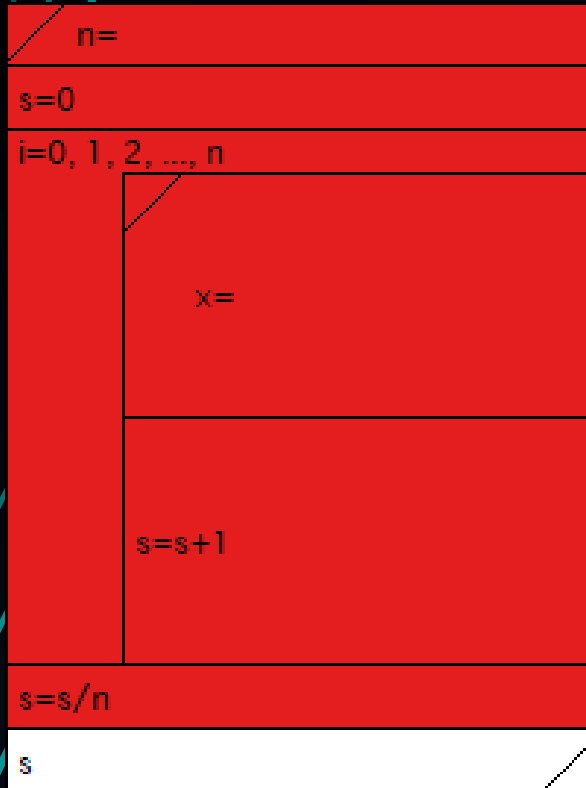
Education Center

- What's New**
New features and updates
- Getting Started**
Use these basic tutorials to create and share diagrams
- Tips and Tricks**
Discover pro tips to become more efficient
- Help Center**
Troubleshoot, learn more, or contact support

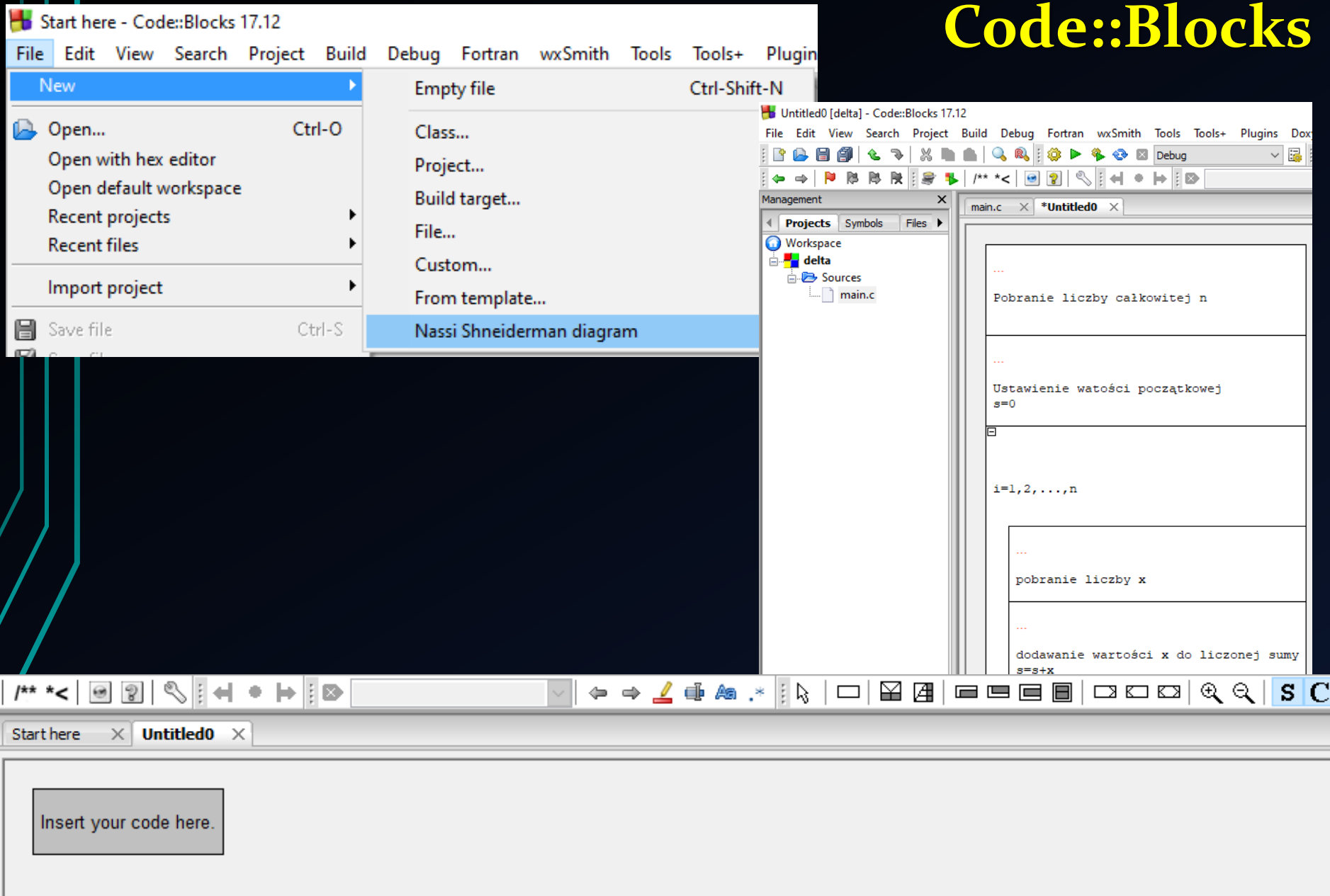




NS Builder



Code::Blocks



- Na podstawie zapisane go program potrafi wygenerować Schemat blokowy NS
- Należy zaznaczyć fragment kodu
- Zaznaczony kod nie może zawierać
 - funkcji użytkownika
 - deklaracji nagłówkowych

```

printf("Program oblicza pierwiastki rownania kwadratowego w postaci  $Ax^2+Bx+C=0$ .\n");

printf("Podaj wspolczynniki rownania A,B,C.\n");

float A,B,C;

printf("A=");
scanf("%f", &A);

(A==0)

printf("B=");
scanf("%f", &B);

printf("C=");
scanf("%f", &C);

float delta;

delta=B*B-4*A*C;

(delta<0)

printf("Delta ujemna, rownanie nie posiada pierwiastkow rzeczywistych");

(delta>0)

float pDeltay,x1,x2;
pDeltay=sqrt(delta);
printf("Delta wynosi: %f\n",delta);
x1=(-B+pDeltay)/(2*A);
x2=(-B-pDeltay)/(2*A);
printf("Rownanie posiada dwa miejsca zerowe:\n");
printf("x1=%f\n",x1);
printf("x2=%f\n",x2);

float x;
x=-B/(2*A);
printf("Rownanie posiada jedno rozwiazanie x=%f",x);

```

Algorytm - przykłady

Zadanie: Obliczanie średniej arytmetycznej dwóch liczb.

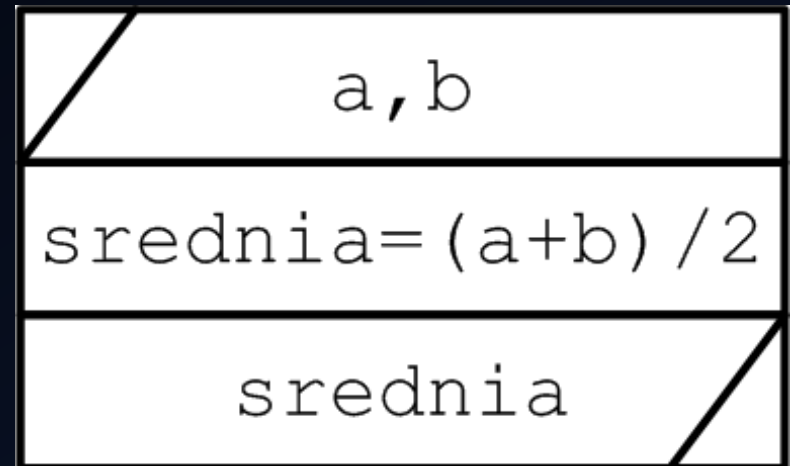
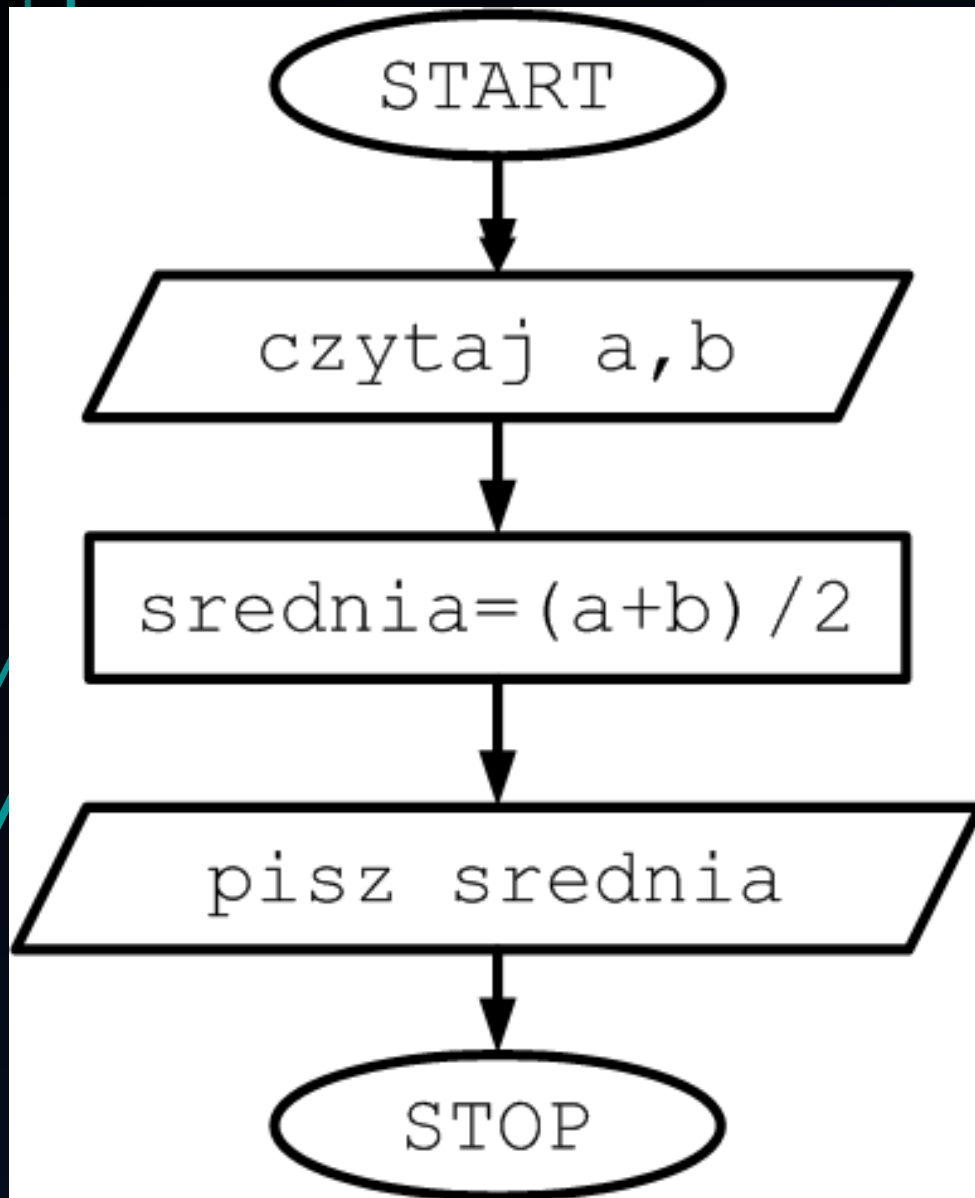
Lista kroków:

1. Wczytaj dwie liczby a i b
2. Dodaj do siebie te liczby i wynik podziel przez 2 – $(a+b)/2$
3. Wyświetl wynik i zakończ algorytm.

Pseudokod:

1. Czytaj a i b
2. $srednia = (a+b) / 2$
3. Pisz $srednia$

Algorytm - schematy



Algorytm - projektowanie

Zadanie: Student w czasie sesji zdaje n egzaminów. Oblicz średnią ocen w sesji.

Lista kroków v1:

1. Wczytaj liczbę egzaminów n
2. Wyzeruj zmienną $s \rightarrow s=0$
3. Wykonaj n razy:
 - a) Wczytaj ocenę x
 - b) Dodaj x do aktualnie obliczonej sumy
 $s \rightarrow s=s+x$
4. Oblicz wartość średnie $\rightarrow s=s/n$
5. Wyświetl wynik s i zakończ algorytm

Pseudokod v1:

1. Czytaj n
2. $s=0$
3. Wykonaj n razy
 - a) Wczytaj x
 - b) $s=s+x$
4. $s=s/n$
5. Pisz s

Algorytm - projektowanie

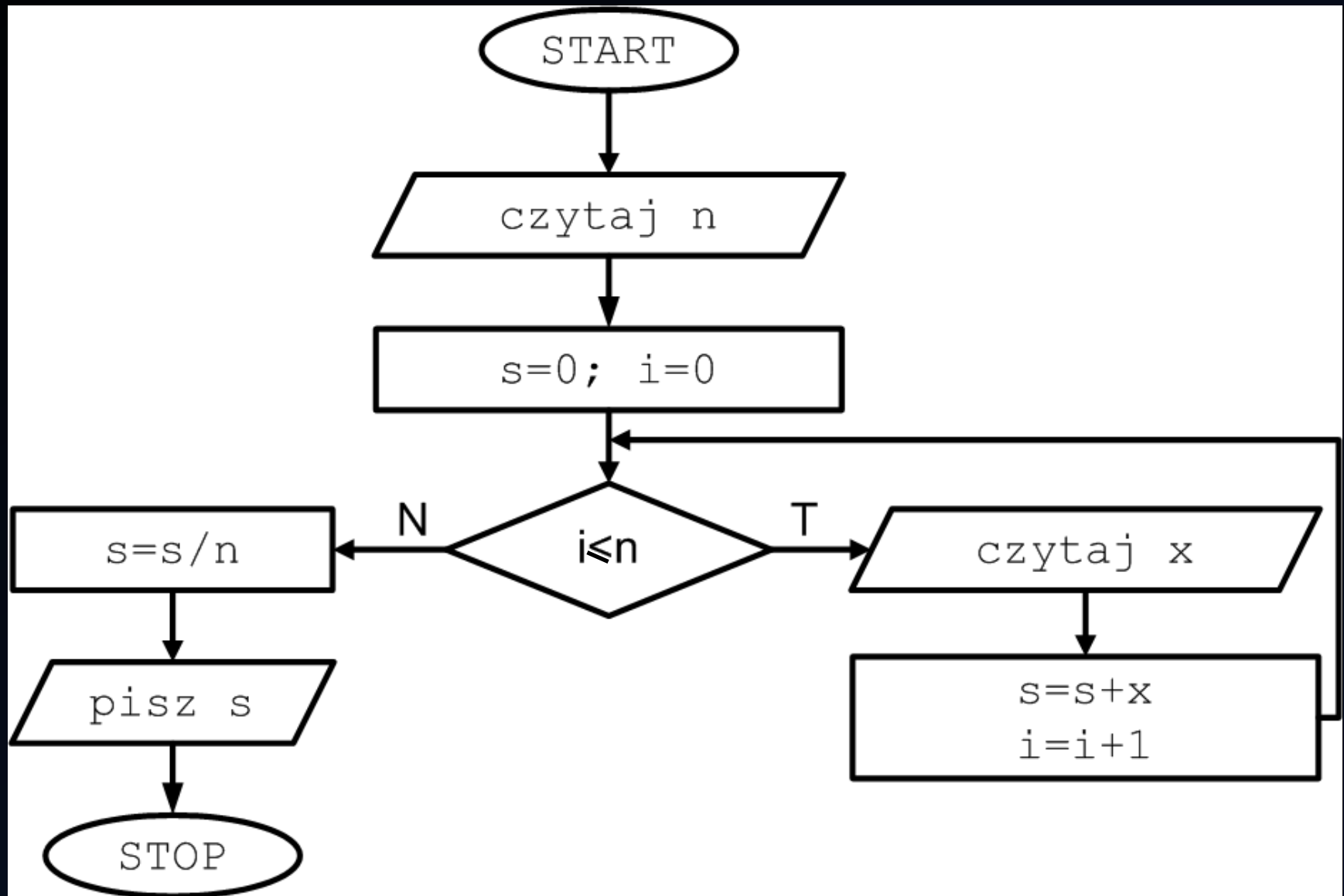
Lista kroków v2:

1. Wczytaj liczbę egzaminów n
2. Wyzeruj zmienną s i $licz \rightarrow s=0; \text{ } licz=0$
3. Powtarzaj:
 - a) Wczytaj ocenę x
 - b) Dodaj x do aktualnie obliczonej sumy
 $s \rightarrow s=s+x$
 - c) Zwiększ $licz$ o 1 $\rightarrow \text{ } licz=licz+1$
aż $licz$ osiągnie wartość $n+1$
4. Oblicz wartość średnie $\rightarrow s=s/n$
5. Wyświetl wynik s i zakończ algorytm

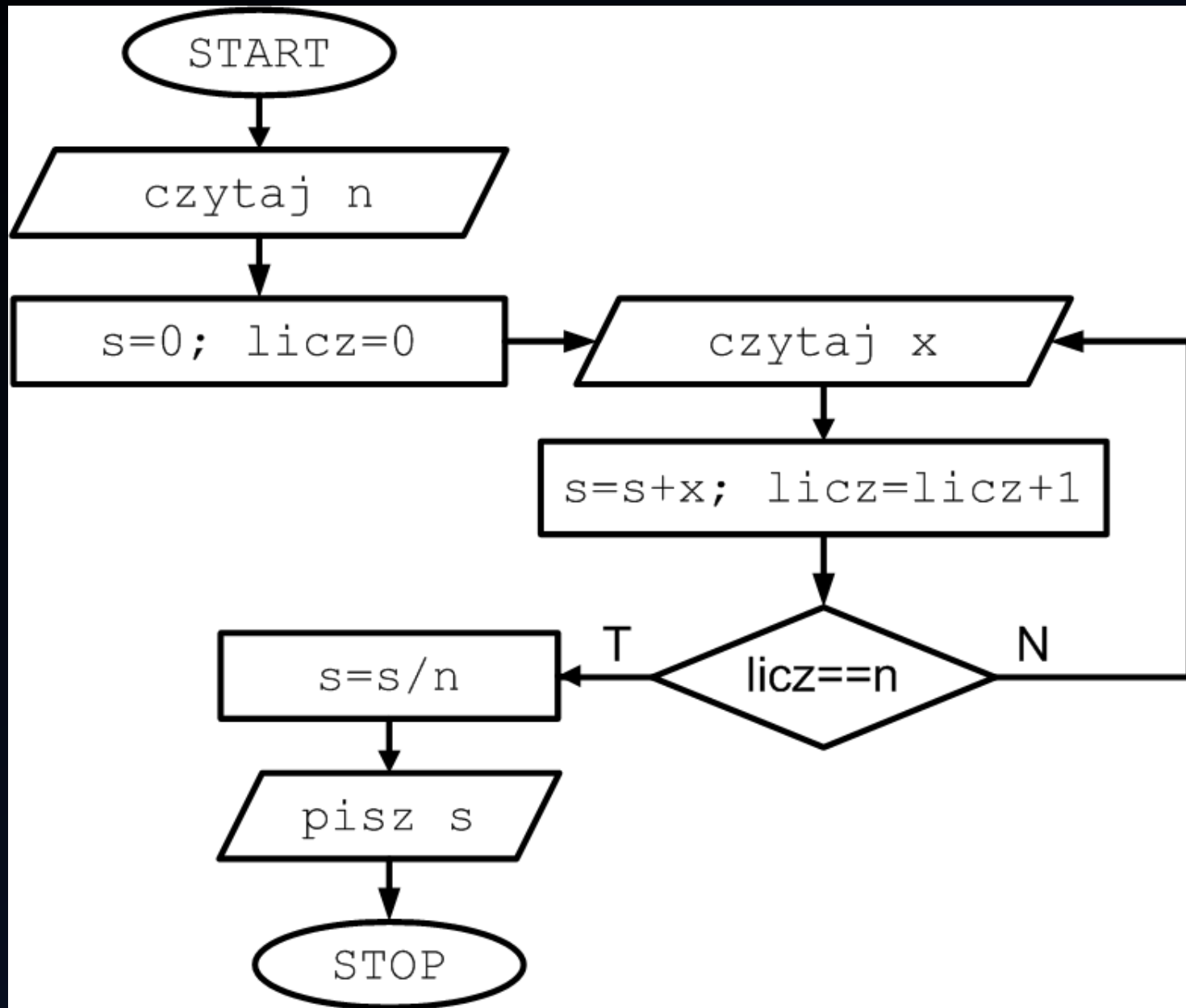
Pseudokod v2:

1. Czytaj n
2. $s=0$ $licz=0$
3. Powtarzaj aż $licz=n+1$:
 - a) czytaj x
 - b) $s=s+x$
 - c) $licz=licz+1$
4. $s=s/n$
5. Pisz s

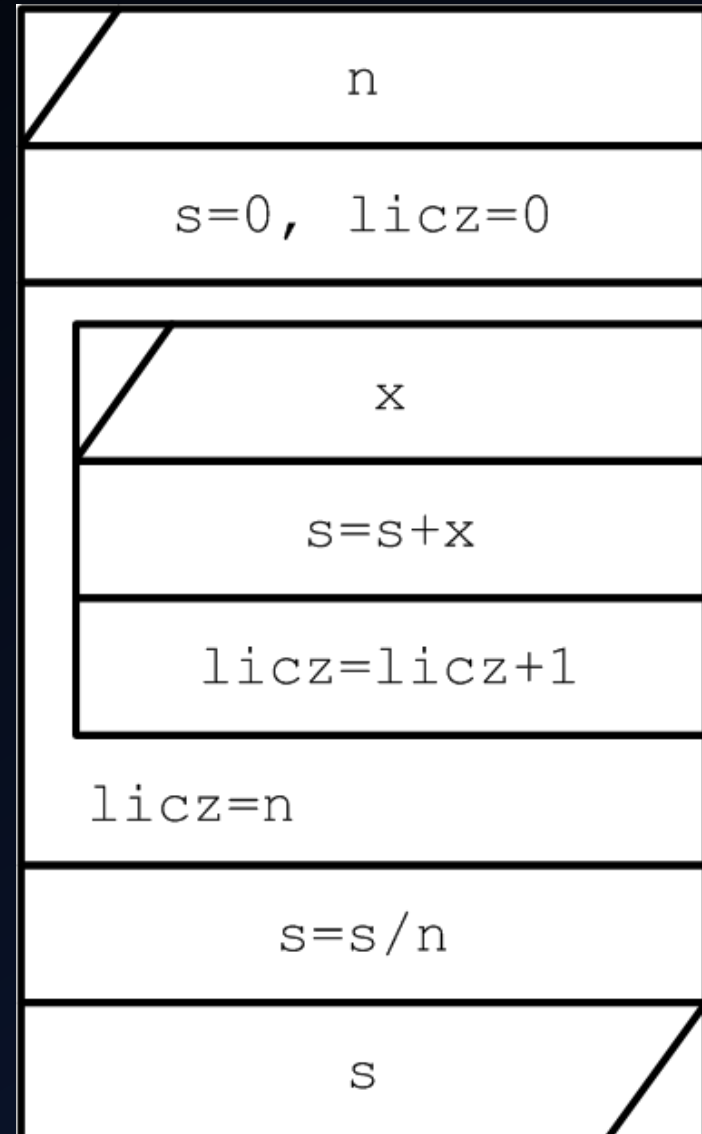
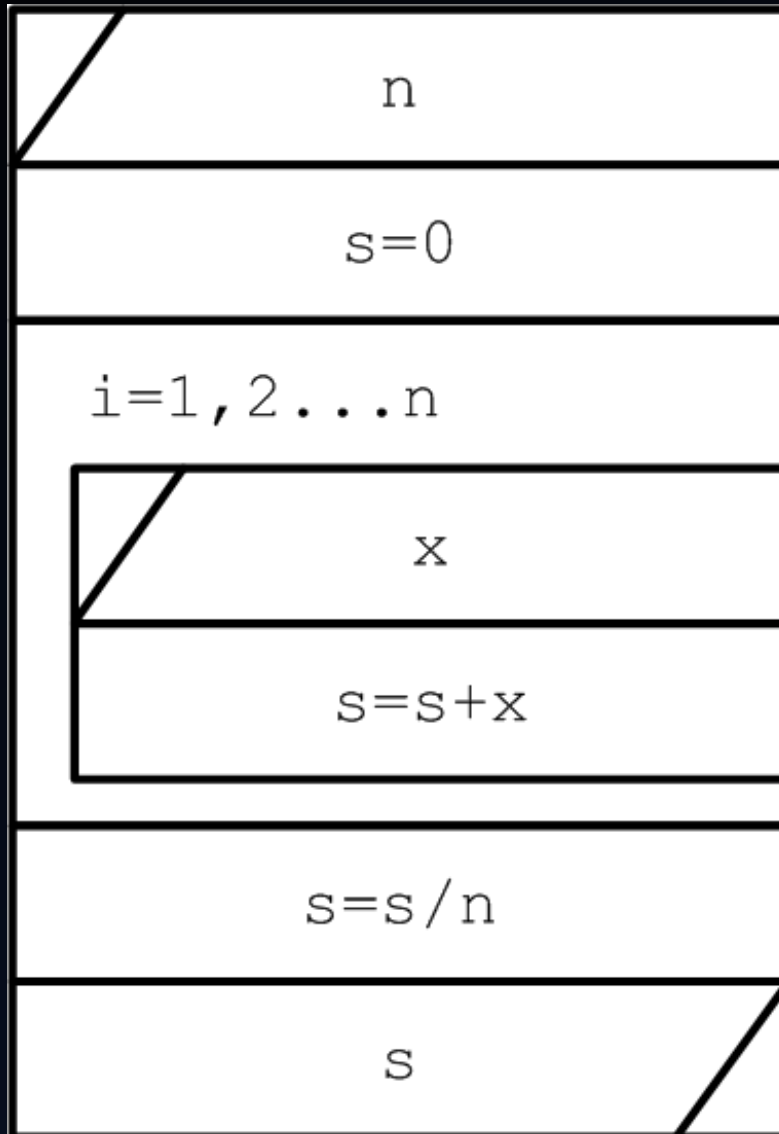
Algorytm – schematy v1



Algorytm – schematy v2



Algorytm – schematy v2



Algorytm – projektowanie

Zadanie: Przy danych współczynnikach równania kwadratowego zbadaj istnienie pierwiastków, a gdy istnieją oblicz je.

Równania:

$$ax^2 + bx + c = 0$$

$$\Delta = b^2 - 4ac$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Algorytm - projektowanie

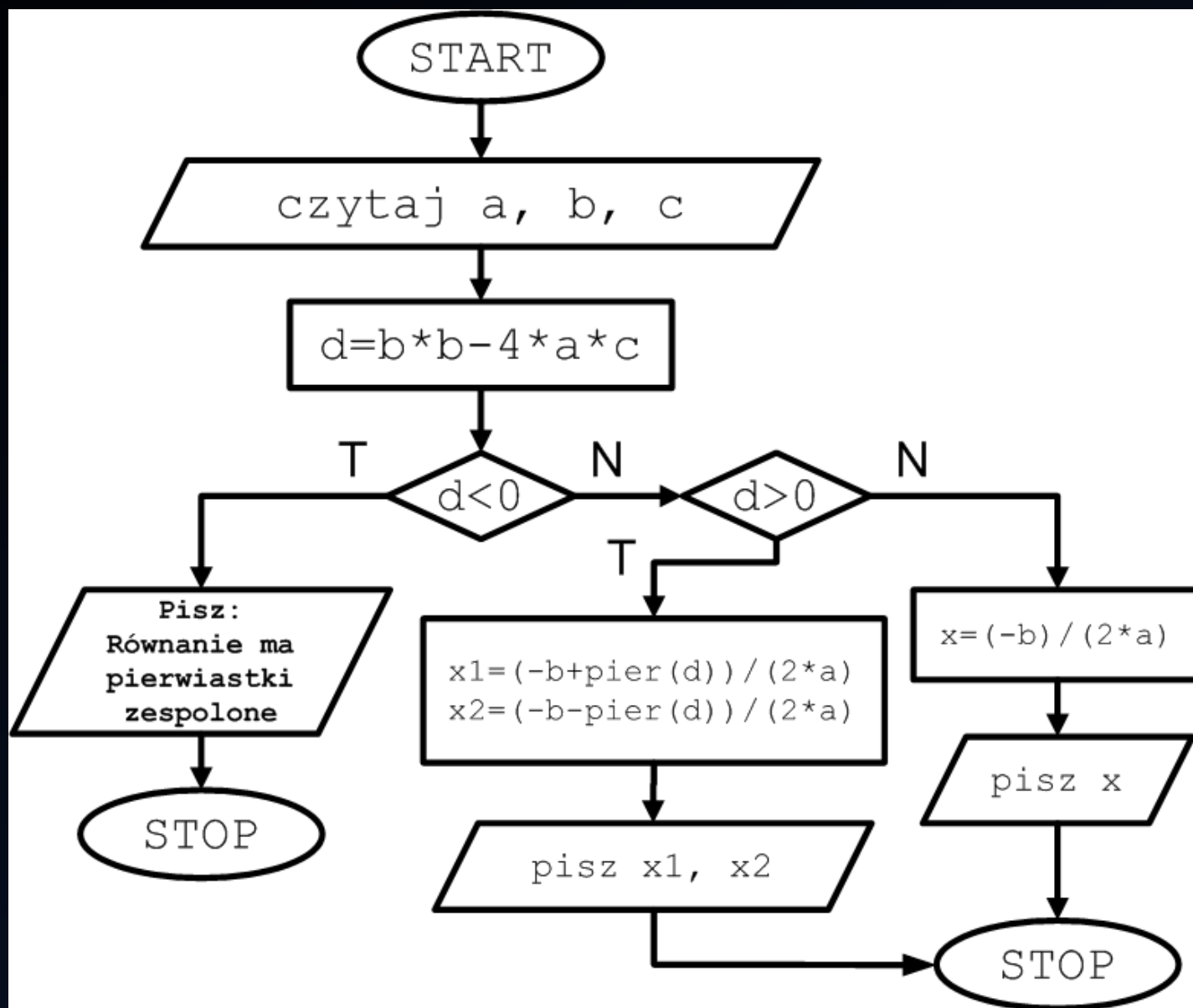
Lista kroków:

1. Wczytaj współczynniki równania a, b, c , ($a \neq 0$)
2. Oblicz delta $d = b^2 - 4 * a * c$
3. Jeśli $d < 0$ to:
 - a) Wyświetl „Równanie ma rozwiązanie zespolone”
 - b) Zakończ algorytm
4. Jeśli $d > 0$ to:
 - a) Oblicz pierwiastki
 $x_1 = (-b + \sqrt{d}) / (2 * a)$ i $x_2 = (-b - \sqrt{d}) / (2 * a)$
 - a) Wyświetl wyniki
 - b) Zakończ algorytm
5. Jeśli $d == 0$ to
 - a) Oblicz pierwiastek $x = -b / (2 * a)$
 - b) Wyświetl wynik
 - c) Zakończ algorytm

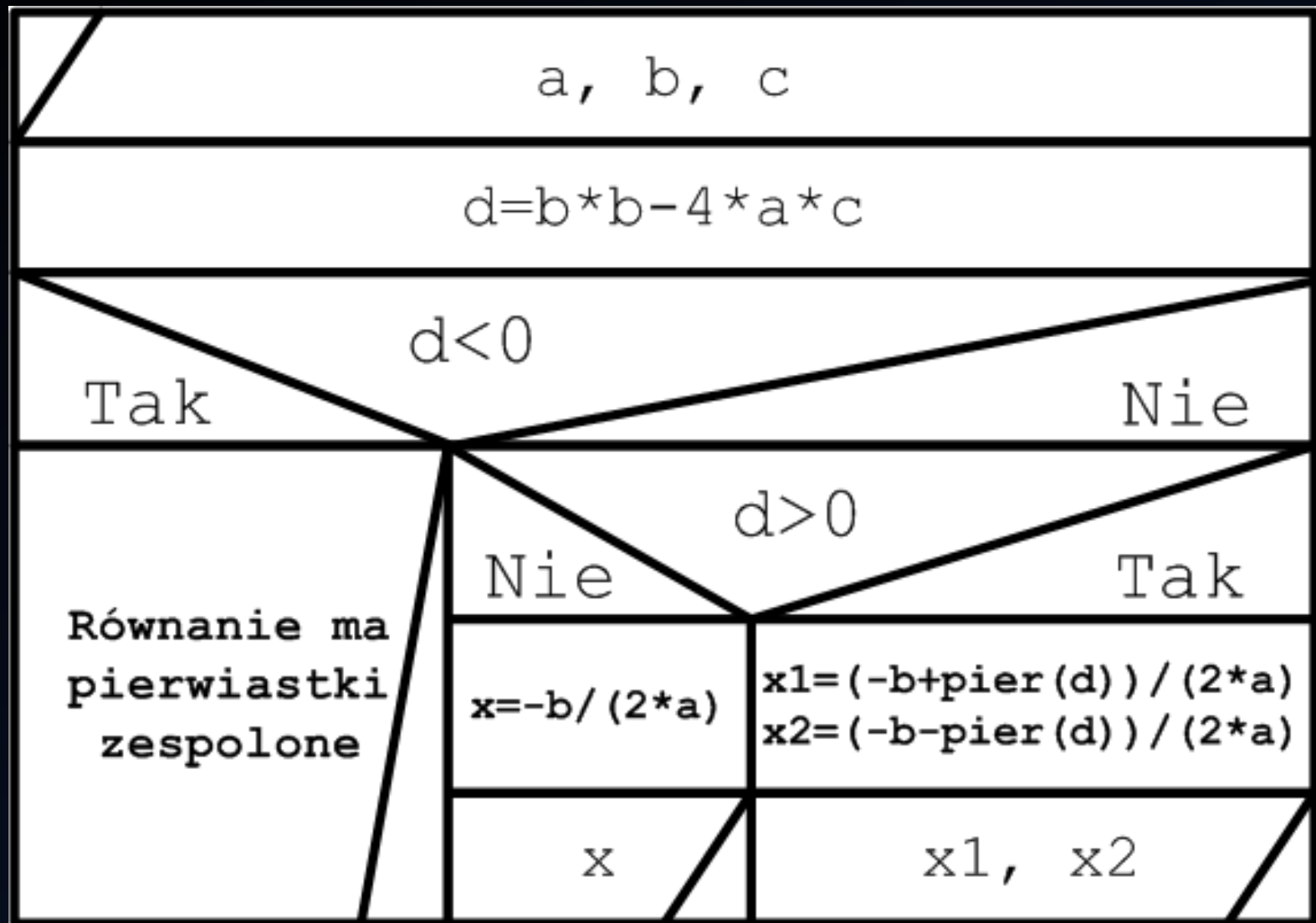
Pseudokod

1. Czytaj a, b, c
2. $d = b^2 - 4 * a * c$
3. Jeśli $d < 0$ to pisz „Równanie ma rozwiązanie zespolone”
jeśli nie to:
4. Jeśli $d > 0$ to:
 - a) Przypisz x_1 i x_2 wartości
 $x_1 = (-b + \text{sqrt}(d)) / (2 * a)$ i $x_2 = (-b - \text{sqrt}(d)) / (2 * a)$
 - a) Pisz x_1 i x_2
jeśli nie to:
 - a) Przypisz x wartość $x = -b / (2 * a)$
 - b) Pisz x

Algorytm - schematy



Algorytm - schematy



- Zaproponować algorytm obliczania największego wspólnego dzielnika NWD
- Algorytm Euklidesa

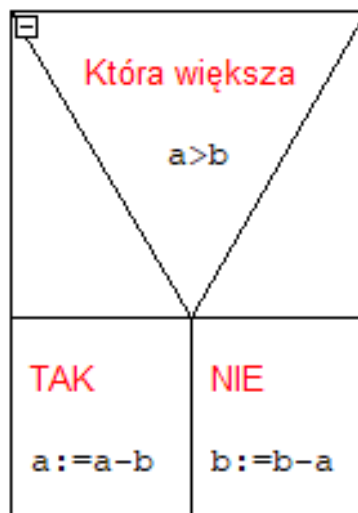
Pobieramy dwie liczby naturalne, od większej z nich odejmujemy mniejszą, a następnie większą liczbę zastępujemy różnicą. Postępujemy tak do momentu, gdy dwie liczby będą równe.

Dane wejściowe

Read (a)
Read (b)

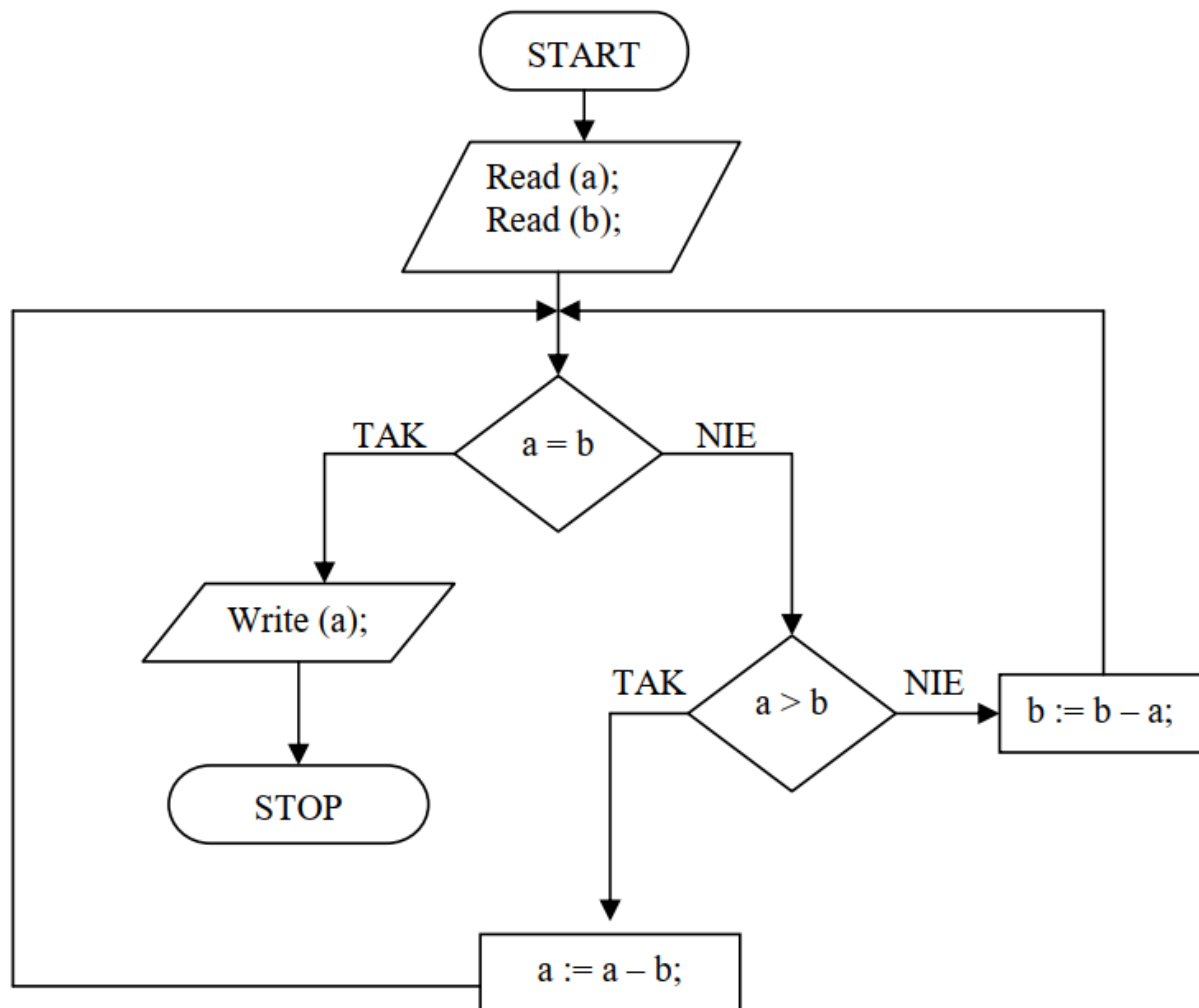
Warunek iteracji

$a \neq b$



Wyświetlenie NWD

write (a)



- Program sprawdza czy wprowadzona liczba jest liczbą pierwszą
- Należy sprawdzić czy kolejne liczby naturalne mniejsze od podanej począwszy od 2 w wyniku dzielenia nie zwracają liczby 0
- Jeżeli liczba całkowita nie dzieli się przez żadną liczbę całkowitą mniejszą lub równą jej pierwiastkowi to jest liczbą pierwszą

Wczytanie liczby

`n`

...

`i:=1`

Warunkowa pętla iteracyjna

`i>sqrt(n)`

warunek liczby pierwszej $n \bmod i = 0$	
NIE <code>i++</code>	TAK <code>n nie jest liczbą pierwszą</code>
Koniec programu <code>return 0</code>	

Koniec programu

`n jest liczbą pierwszą`

