ZAJĘCIA NR 5

Algorytm Euklidesa – znajdywanie NWD (=największy wspólny dzielnik)

Niech n, m ∈ N.

1) Jeśli m = n, to NWD jest liczbą m; koniec obliczeń.
2) Jeśli m > n, to nową wartość m przyjmij m-n; w przeciwnym przypadku za nową wartość n przyjmij n-m.
3) Przejdź do punktu 1) .

W wyniku działania tego algorytmu (w jego trakcie) zmienia ulegają wartości m i n. Taka operacja polegająca na zmianie wartości nosi nazwę operacji podstawienia (lub przypisania).

Przykład:

m	n
75	63
12	
	51
	39
	27
	15
	3
9	
6	
3	
NWD = 3	

m	n
75	63
12	51
9	39
6	27
3	15
	3
NWD = 3	

Obydwie te tabele przedstawiają ten sam przykład (obliczenie NWD liczb 75 i 63). Jednak lewa tabela – w stosunku do prawej – dodatkowo uwzględnia kolejność wykonywaniach operacji.

Operacje podstawiania opisujemy według następującego wzorca: N := W, co czytamy: za N podstaw wartość W, gdzie N jest nazwą obiektu, którego wartość chcemy zmienić, a W jest wyrażeniem, którego wartość ma się stać nową wartością N, a znak (a właściwie: dwuznak) „:=” nazywamy symbolem podstawiania.

Teraz algorytm Euklidesa możemy więc zapisać następujący sposób:

Niech n, m ∈ N.

1) Jeśli m = n, to NWD jest liczbą m; koniec obliczeń.
2) Jeśli m > n, to m := m-n; w przeciwnym przypadku n := n-m.
3) Przejdź do punktu 1) .

W informatyce zamiast pisań E := mc^2 pisze się: E:=m*c↑2 (chodzi o liniowość i jednoznaczność zapisu).

i := i + 1 – za i podstaw dotychczasową wartość i powiększoną o 1. Jest to zapis jak najbardziej poprawny. Oczywiście i = i + 1 jest zapisem niepoprawnym. Przedstawia bowiem równanie (matematyczne, a nie operację podstawiania w informatyce), w którym po redukcji i

otrzymujemy: 0 = 1, a więc sprzeczność). Tak określone i „robi za” licznik kolejnych kroków pętli czy zliczania ilości zajść pewnej sytuacji.

Przedstawienie algorytmu przy pomocy schematu blokowego.

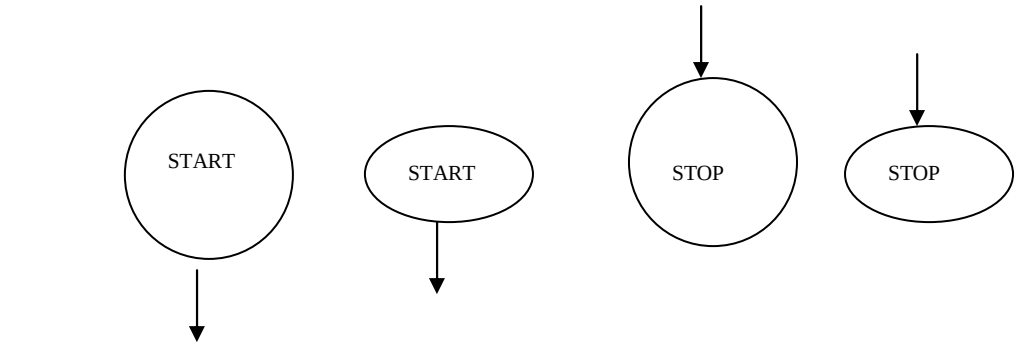
Sieć działań = schemat blokowy, zajmuje się tylko opisem czynności.

Def
Schemat blokowy jest to graficzne przedstawienie ciągu wykonywanych operacji uwzględniających ich wzajemna kolejność. Składa się on:
- z figur geometrycznych – zwanych skrzynkami – określających wykonywane operacje
- i łączących je linii skierowanych – tzw. strzałek – wskazujących kolejność operacji.

W zasadzie w jednej skrzynce powinna być zapisana tylko jedna operacja.

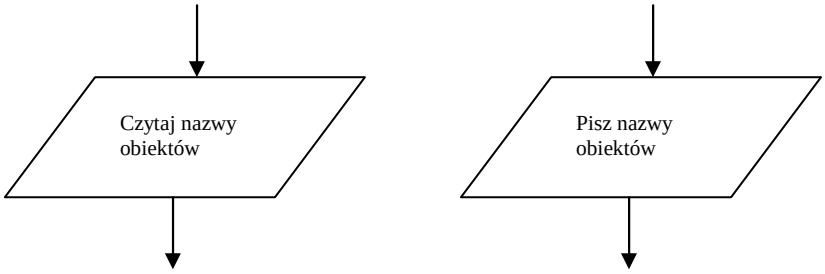
Polska norma „PN 72 E01226” normalizuje rodzaje skrzynek.
Proponuje się stosować następujące skrzynki:

1) Skrzynki graniczne:



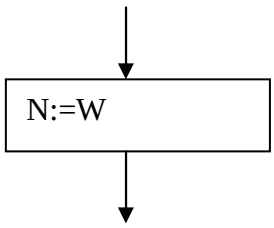
Określają one początek i koniec schematu blokowego. Skrzynki wejścia mają 1 wyjście (i brak wejść), a skrzynki wyjścia – 1 wejście (i brak wyjść).

2) Skrzynki wprowadzania i wyprowadzania danych (skrzynki WE / WY)



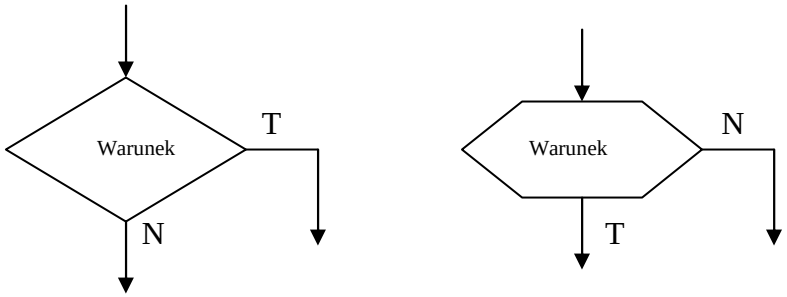
Czytaniu zwykle podlegają dane, a pisaniu wyniki.
Skrzynki te mają po 1 wejściu i po 1 wyjściu.

3) Skrzynka operacyjna



Ma ona 1 wejście i 1 wyjście.

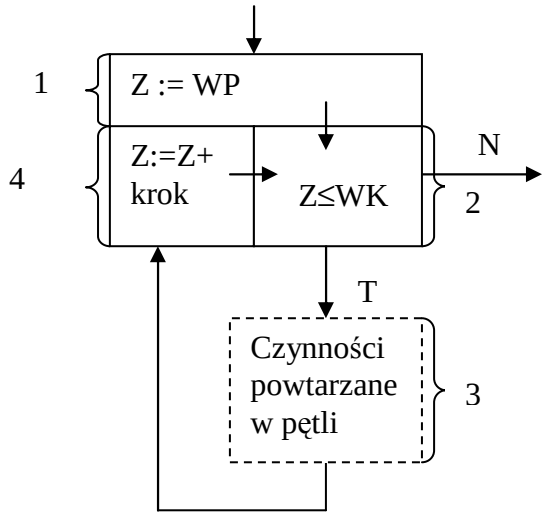
4) Skrzynka warunkowa



Ma ona 1 wejście i 2 wyjścia, przy czym to, do którego się skierujemy, jest zależne od tego, czy zaszedł warunek (T) czy nie (N).
Krzynka ta umożliwia więc nam zapis instrukcji warunkowej. Pozwala na rozgałęzienie obliczeń.

5) Skrzynki łącznikowe (mają charakter pomocniczy; stosujemy gdy chcemy utożsamić ze sobą dane punkty schematy, np. 2 różnych stron).

6) skrzynka pętli

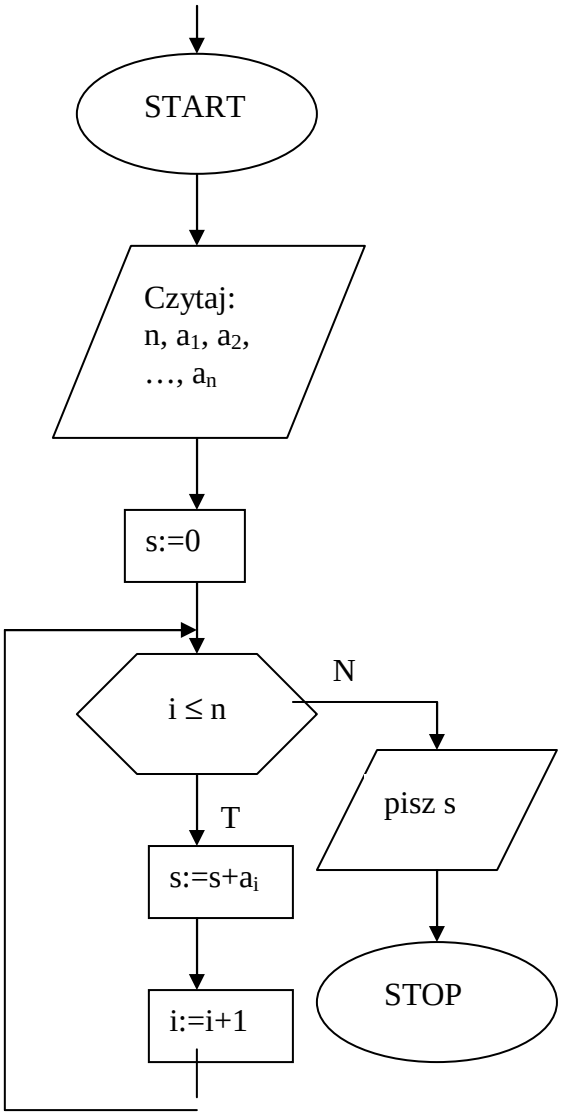


Gdzie:
WP – wartość początkowa
WK – wartość końcowa
Krok – jest wartości o którą każdorazowo zwiększamy wartość obiektu o nazwie Z pełniącego rolę obiektu sterującego wykonaniem pętli.
Dzięki tak skonstruowanej pętli można powtarzać operacji w niej zamieszczone (tu: w zaznaczonej linią przerywaną skrzynce).

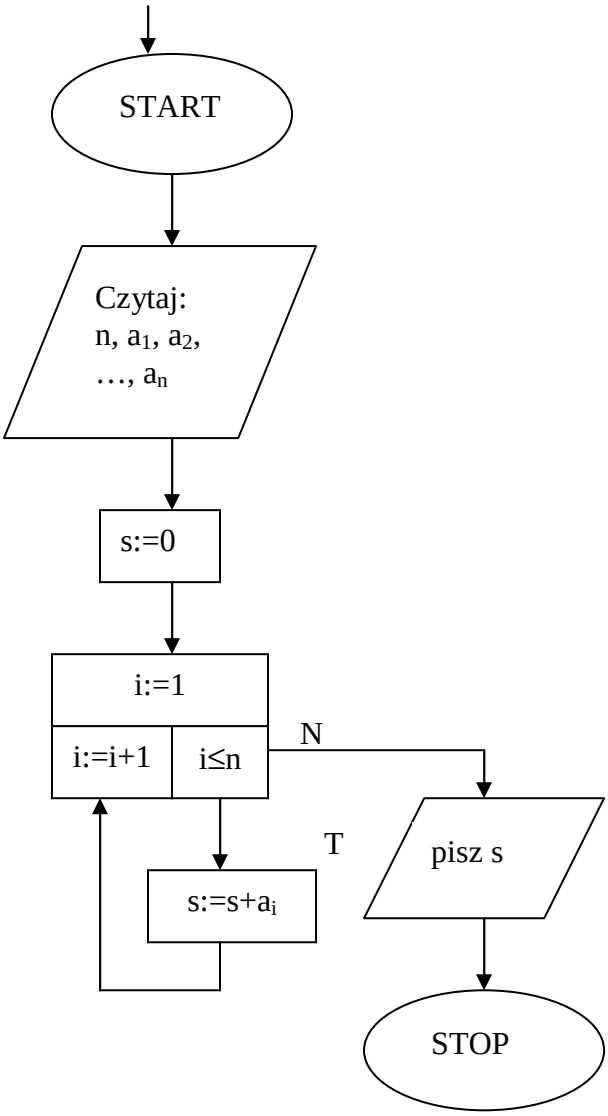
Części skrzynki:
1 – część inicjująca wartość obiektu sterującego,
2 – część sprawdzająca warunek zakończenia pętli,
3 – zestaw czynności do powtarzania,
4 – część modyfikująca obiekt sterujący.

ZADANIE:
Dane są liczby: $a_1, a_2, \ldots, a_n$. Należy zesumować te liczby. Napisać algorytm.

I rozwiązanie
(bez wykorzystania skrzynki pętli):



II rozwiązanie
(z wykorzystaniem skrzynki pętli):



Definicja algorytmu zawiera pojęcie obiektu i pojęcia działania (czyli czynności). Zauważmy, że każde działanie wymaga istnienia pewnego obiektu, a nad którym to działanie przebiega, i po zmianie stanu którego można sądzić o fakcie działania. Aby opisać obiekty i działania musimy mieć do dyspozycji pewien język nazywany językiem programowania. Opis obiektu w języku programowania będziemy nazywać deklaracją, a opis czynności (działania) nazywać będziemy instrukcją. Formalnie zapisany algorytm w języku programowania tworzył będzie program.

Konstruowanie algorytmów

Jeśli działania rozłożyć można na części składowe, to taki zestaw działań nazywa się procesem lub obliczeniem. Jeśli części składowe występują kolejno jedna po drugiej, to mamy proces sekwencyjny. To, co wykonuje działania, zgodnie z ustalonymi instrukcjami, nazywać będziemy procesorem. Wymagane jest przy tym, aby programista opisał przebieg obliczeń na poziomie rozumianym przez procesor. Programista powinien więc znać typy instrukcji, które zdolny jest wykonywać procesor i pisać instrukcje te w pełnej zgodności ze stosowanym językiem.

Algorytmy charakteryzuje następujące 5 elementów:

1) Każdy z tych algorytmów charakteryzuje się skończonością (działanie algorytmu powinno zakończyć się po skończonej, sensownej liczbie kroków),
2) Algorytmy powinny charakteryzować się pełną określonością (tzn. każdy krok algorytmu powinien być dokładnie określony, a wykonywane czynności powinny być ścisłe i jednoznacznie wykonywane w każdym rozpatrywanym przypadku; działania powinny prowadzić do określonego wyniku – pośredniego lub końcowego),
3) Algorytmy powinny posiadać tzw. wejście, tzn. pewien zestaw danych wejściowych umożliwiających wykonywanie obliczeń,
4) Algorytmy powinny posiadać tzw. wyjście, tzn. zestaw danych wyjściowych będących wynikami przeprowadzonych operacji,
5) Algorytmy powinny być efektywne, tzn. wszystkie operacje jakie zawarto w algorytmie powinny być dostatecznie proste.

Poprawność algorytmu

Algorytm jest poprawny, gdy uzyskujemy w wyniku jego pracy poprawny wynik (przy wszelkich sytuacjach!).

Załóżmy, że nasz mechanizm sprawdzania poprawności działania programu umożliwia zapisywanie liczb od 1 do 10.000.000. Kombinacji par jest wówczas $(10^7)^2$. Załóżmy, że jedną operację wykonuje przez 10^{-3} sek. Jest to: $(10^7)^2 * 10^{-3}$ sek. = 10^{11} sek. ≈ 10^6 dni

Tak więc dowieść poprawności metodą sprawdzania nie można, ale żeby dowieść błędności algorytmu – wystarczy znaleźć chociaż 1 przypadek, gdy „siada”.

W oparciu o logikę i informatykę – rozwinęła się nauka badająca poprawność programu: najpierw bada się poprawność instrukcji, potem operacji, ciągów operacji, a wreszcie całego programu.

Np. w algorytmie Euklidesa wykorzystuje się następujące własności matematyczne:

1) NWD(m, n) = NWD(n, m)
2) NWD(m, m) = m
3) NWD(m, n) = NWD(m-n, n), gdy m>n
4) NWD(m, n) = NWD(m, n-m), gdy n>m

Powstała tez oddzielna dziedzina wiedzy polegająca na udowadnianiu twierdzeń matematycznych w oparciu o wynik działania programu komputerowego (któ®y oczywiście oparty jest na pewnym algorytmie). W ten sposób udowodniono np. zagadnienia kolorowania mapy politycznej za pomocą jedynie 4 barw.

CDN…