

Co to jest algorytm ?

Na lekcjach matematyki czy fizyki często słyszymy zdanie "rozwiąż zadanie. Większość ich rozwiązujemy według pewnych schematów. Najpierw wypisujemy dane i zastanawiamy się do jakiego celu dążymy, a więc jaki ma być wynik. Następnie wypisujemy wzory łączące dane z szukanymi bądź twierdzenia, które można zastosować.

Przed koniecznością rozwiązywania zadań stajemy również poza szkołą. Robimy to w każdej innej dziedzinie naszego życia, na przykład:

- telefonując do koleżanki i zapraszając ją na ciasto
- gotując jajko na twardo
- kupując piłkę w sklepie
- znajdując najniższego ucznia w klasie
- pisząc wypracowanie

Również przy takim typie zadań musimy określić dane i warunki, które muszą one spełniać. Formułujemy także wynik, który pragniemy uzyskać.

Algorytm to przepis rozwiązania zadania, zawierający opis danych wraz z opisem czynności, które należy wykonać z tymi danymi, aby osiągnąć zamierzony cel.

Etapy rozwiązywania problemów

W procesie rozwiązywania każdego zadania możemy wyróżnić pewne etapy, które nas do niego prowadzą.

Etapy rozwiązywania problemów:

1. Sformułowanie zadania
2. Określenie danych wejściowych
3. Określenie celu, czyli wyniku
4. Poszukiwanie metody rozwiązania, czyli algorytmu
5. Przedstawienie algorytmu w postaci:
 - opisu słownego
 - listy kroków
 - schematu blokowego
 - języka programowania
6. Analiza poprawności rozwiązania
7. Testowanie rozwiązania dla różnych danych. Ocena efektywności przyjętej metody.

Działania niealgorytmiczne

Zastanów się: Czy wszystkie działania są algorytmiczne?

Czy dla każdego zadania można skonstruować algorytm? Czy rozwiązanie każdego zadania polega na wykonywaniu jednoznacznie opisanych, ściśle określonych czynności? Oczywiście, że nie. Istnieją zadania, których realizacji nie można ująć w ramy jakiegoś planu działania. Taki charakter ma np. każda twórczość artystyczna. Konieczna jest do tego wyobraźnia i twórcze działanie, a na to nie ma przepisu.

Budowa algorytmów

Algorytmy powinny być tak przedstawiane, aby było możliwe ich jednoznaczne odczytanie i zastosowanie. Niektóre algorytmy można opisać w języku potocznym, zwłaszcza wtedy, gdy jego wykonawcą ma być człowiek (w informatyce zajmujemy się opracowywaniem algorytmów, których wykonanie powierzamy komputerom).

Z czego składa się algorytm?

Zawiera on opis danych, opis wyników oraz plan działania, czyli przetworzenia danych. Plan ten można przedstawić w postaci ciągu czynności, które muszą być wykonane w określonej kolejności. Opis czynności występujących w algorytmie nazywamy instrukcjami.

Algorytm liniowy

Ćwiczenie

1

Opracuj algorytm gotowania jajka na miękko.

Zastanów się jak ugotować jajko na miękko. Na początku opracowywania algorytmu przyjmijmy założenie, że używamy kuchenki gazowej, posiadamy garnek i wodę. Oczywiście niezbędne jest też samo jajko. Zakładamy również, że nic nie utrudni samej czynności, to znaczy np. w trakcie gotowania nie zostaniemy pozbawieni dopływu gazu, czy też osoba nie wie co to garnek.

Algorytm ten ma postać:

1. Wlać do garnka zimną wodę.
2. Zapalić gaz.

3. Gotować wodę do wrzenia.
4. Włożyć jajko.
5. Odczekać trzy minuty.
6. Zgasić gaz.
7. Wyjąć jajko

Ma on prostą postać. Składa się z ciągu instrukcji, które są wykonywane jedna po drugiej w kolejności, jaka wynika z ich następstwa w zapisie algorytmu. Taki algorytm nosi nazwę algorytmu liniowego (sekwencyjnego).

Ćwiczenie

2

Opracuj algorytm telefonicznego zaproszenia koleżanki na ciasto. Numer telefonu : 616-22-22.

Wiesz już jak wyglądał poprzedni algorytm oraz co to jest algorytm liniowy. W tym przypadku zrobimy założenie, że nic nie utrudni połączenia (a więc już za pierwszym razem zostaniemy połączeni z wybraną osobą), a wykonawca czynności zna cyfry i wie jak wybiera się je z tarczy telefonu.

Oto jak wygląda algorytm (wersja 1):

1. Podnieś słuchawkę.
2. Wybierz cyfrę 6.
3. Wybierz cyfrę 1.
4. Wybierz cyfrę 6.
5. Wybierz cyfrę 2.
6. Wybierz cyfrę 2.
7. Wybierz cyfrę 2.
8. Wybierz cyfrę 2.
9. Przekaż informacje.
10. Odlóż słuchawkę.

Algorytmy liniowe mają opisy składające się z kroków, które nie zależą od żadnych warunków i są wykonywane w zapisanej kolejności. Istnieją jednak sytuacje, w których dalsze postępowanie w algorytmie zależy od spełnienia, bądź nie, określonych warunków. Czasami musimy powtórzyć pewne kroki algorytmu kilka razy.

Instrukcja warunkowa

W rzeczywistości, większość algorytmów ma bardziej rozbudowaną strukturę. Często występują w nich instrukcje, których wykonanie uzależnione jest od spełnienia pewnego warunku lub też spełnienie pewnego warunku powoduje wykonanie jednej instrukcji, a niespełnienie go -innej. Taką instrukcję nazywamy instrukcją warunkową. Działa ona według jednego z dwóch przedstawionych schematów:

Jeśli spełniony jest warunek W , wykonaj instrukcję A .

Jeśli spełniony jest warunek W , to wykonaj instrukcję A ; w przeciwnym razie wykonaj instrukcję B .

Instrukcja A i B opisuje jedną instrukcję lub instrukcję składającą się z ciągu instrukcji wykonywanych sekwencyjnie. Instrukcja warunkowa pozwala dokonać wyboru jednej z dwóch dalszych dróg wykonania algorytmu.

Ćwiczenie

2

Popraw opracowany wcześniej algorytm tak, aby uwzględniał sytuację, gdy po wybraniu numeru jest on zajęty lub połączenie okazało się błędne.

Kiedy slychać sygnał zajętości numeru, a więc nie udało się uzyskać połączenia trzeba Odłożyć słuchawkę. Tak samo postępujemy, gdy nawiązane zostało połączenie z innym Numerem.

Aby zrealizować taką sytuację zastosujemy instrukcję warunkową. Zrobimy to po to, aby opisać czynności, powinny być wykonane wtedy kiedy zostało nawiązane poprawne połączenie, jak również nie zostało nawiązane. Zauważ, że wtedy wykonawca znajdzie się w punkcie wyjścia, czyli jakby w ogóle nie podjął próby telefonowania.

Algorytm może mieć postać taką (wersja 2):

1. Podnieś słuchawkę.
2. Wybierz cyfrę 6.
3. Wybierz cyfrę 1.
4. Wybierz cyfrę 6.
5. Wybierz cyfrę 2.
6. Wybierz cyfrę 2.
7. Wybierz cyfrę 2.
8. Wybierz cyfrę 2.
9. Czy połączyłeś się z koleżanką ?
 - A. Jeśli TAK, to przejdź do kroku 10.
 - B. Jeśli NIE, to przejdź do kroku 11.
10. Zaproś koleżankę.
11. Odłóż słuchawkę.

Instrukcja iteracyjna

Instrukcja iteracyjna - ze znaną ilością powtórzeń

Przyjrzyj się uważnie algorytmowi. Zauważyłeś, że istnieją tu powtarzające się instrukcje, aż czterokrotnie występuje "Wybierz cyfrę 2". Takie wielokrotne powtarzanie niektórych instrukcji jest cechą charakterystyczną wielu algorytmów, jednak nie zawsze (tak jak w tym algorytmie) możemy określić dokładnie liczbę powtórzeń. Może ona zależeć od spełnienia pewnych warunków. Wielokrotne

powtarzanie instrukcji umożliwia instrukcje iteracyjne (pętle) . Działa ona według schematu:

Wykonuj instrukcję A dokładnie n razy.

Ćwiczenie

Popraw opracowany wcześniej algorytm tak, aby sekwencję jednakowych czynności zastąpić pętlą.

Oto algorytm (wersja 3):

1. Podnieś słuchawkę.
2. Wybierz cyfrę 6.
3. Wybierz cyfrę 1.
4. Wybierz cyfrę 6.
5. Wykonaj czynność cztery razy
 - A. Wybierz cyfrę 2.
6. Czy połączyłeś się z koleżanką ?
 - A. Jeśli tak, to przejdź do kroku 7.
 - B. Jeśli nie, to przejdź do kroku 8.
7. Zaproś koleżankę.
8. Odłóż słuchawkę.

Działanie algorytmu może zakończyć się na dwa różne sposoby, albo uzyskamy połączenie z koleżanką i zaprosimy ją na ciasto, albo nie połączymy się i wykonanie algorytmu ograniczy się tylko do wykręcenia numeru. Jak widzisz zastosowanie pętli zmienia sposób zapisu algorytmu, nie zmienia się jednak jego działanie.

Instrukcja iteracyjna - ze spełnieniem warunku

Ćwiczenie

Uwzględnij w opracowanym wcześniej algorytmie przypadek braku połączenia lub nawiązanie nieprawidłowego połączenia.

W poprzednim algorytmie w przypadku uzyskania nieprawidłowego połączenia bądź jego braku, przechodziliśmy do ostatniego kroku , w którym odkładaliśmy słuchawkę. Kończyło się Działanie algorytmu. W tej sytuacji powinniśmy rozpocząć raz jeszcze jego wykonywanie, nie zostało to jednak opisane w konstrukcji. Rozbudujemy teraz algorytm, tak by powtarzano wybieranie numeru aż do uzyskania połączenia. Dopiszemy w tym celu polecenie będące drugim rodzajem instrukcji iteracyjnej: **Powtarzaj wykonywanie instrukcji A aż do spełnienia warunku W.** Czym jest instrukcja A, czym warunek W ?
Instrukcja A - podniesienie słuchawki, wybranie numeru
Warunek W - uzyskanie połączenia z wybranym numerem

Algorytm (wersja 4):

1. Czy słuchawka jest odłożona ?
 - A. Jeśli tak, to przejdź do kroku 2.

- B. Jeśli nie, to odłóż słuchawkę.**
- 2. Podnieś słuchawkę.**
- 3. Wybierz cyfrę 6.**
- 4. Wybierz cyfrę 1.**
- 5. Wybierz cyfrę 6.**
- 6. Wykonaj czynność cztery razy**
 - A. Wybierz cyfrę 2.**
- 7. Czy połączyłeś się z koleżanką ?**
 - A. Jeśli tak, to przejdź do kroku 8.**
 - B. Jeśli nie, to przejdź do kroku 9.**
- 8. Zaproś koleżankę.**
- 9. Odłóż słuchawkę.**

W algorytmie telefonicznego zaproszenia koleżanki dodaliśmy instrukcję, która uwzględniła, że po nieudanej próbie uzyskanie połączenia należy odłożyć słuchawkę, a dopiero później ponownie ją podnieść i spróbować raz jeszcze. W przypadku pierwszej próby nawiązania połączenia również sprawdzamy odłożenie słuchawki.

Instrukcja iteracyjna

W naszym algorytmie sprawdzamy warunek uzyskania połączenia z koleżanką (krok 4). Czasami zdarza się jednak, że już po podniesieniu słuchawki słychać, że linia jest zajęta. Należałoby wtedy odłożyć słuchawkę i ponownie ją podnieść. Jeśli okazałoby się, że nadal słychać w niej sygnał zajętości linii czynność należałoby powtórzyć. Musielibyśmy wykonywać te czynności dopóki linia nie byłaby "czysta". W takim przypadku stosujemy instrukcję, która działa według schematu:

Dopóki warunek W jest spełniony, wykonuj instrukcję A.

Algorytm (w wersji 5) wygląda tak :

- 1. Czy słuchawka jest odłożona ?**
 - A. Jeśli tak, to przejdź do kroku 2.**
 - B. Jeśli nie, to odłóż słuchawkę.**
- 2. Podnieś słuchawkę.**
- 3. Czy linia jest zajęta ?**
 - A. Jeśli Tak, to:**
 - a. Odłóż słuchawkę.**
 - b. Podnieś słuchawkę.**
 - c. Przejdź do kroku 3.**
 - B. Jeśli Nie, to przejdź do kroku 4.**
- 4. Wybierz cyfrę 6.**
- 5. Wybierz cyfrę 1.**
- 6. Wybierz cyfrę 6.**
- 7. Wykonaj czynność cztery razy**
 - A. Wybierz cyfrę 2.**
- 8. Czy połączyłeś się z koleżanką ?**
 - A. Jeśli tak, to przejdź do kroku 9.**
 - B. Jeśli nie, to przejdź do kroku 1.**
- 9. Zaproś koleżankę.**

10. Odlóż sluchawkę.

Jak widzimy w kroku 3 występuje pętla, w której sprawdzamy najpierw warunek, a dopiero potem wykonywana jest instrukcja opisana przez kroki a, b. Jeśli warunek nie jest spełniony, to instrukcja nie zostanie wykonana ani razu.

W informatyce możemy realizować szczególny rodzaj powtórzeń bez konieczności stosowania pętli. Jest to technika rekurencji .

Sposoby zapisywania algorytmów

Algorytmy powinny być tak przedstawiane, aby było możliwe ich jednoznaczne odczytanie i zastosowanie. Można prezentować je poprzez:

1. Zapis w postaci ciągu kroków (języka potocznego)
2. Zapis w postaci graficznej - schematy blokowe
3. Zapis w języku symbolicznym
4. Zapis w języku programowania

Algorytm wyrażony w jakimś języku programowania nazywa się programem.

Schematy blokowe



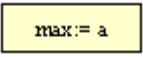

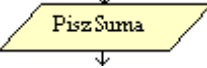
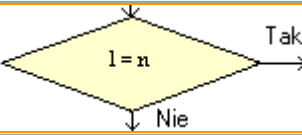
Do tej pory do opisu algorytmów używaliśmy języka potocznego. Były to jednak algorytmy proste, które ma wykonywać człowiek. W przypadku algorytmów skomplikowanych ten zapis będzie nieczytelny, nie sprawdzi się. Dlatego teraz poznamy nowy, bardziej przejrzysty sposób - schematy blokowe .

Schemat blokowy to graficzny zapis algorytmu rozwiązania zadania, przedstawiający opis i kolejność wykonywania czynności realizujących dany algorytm.

W schemacie blokowym poszczególne operacje przedstawione są za pomocą odpowiednio połączonych skrzynek (klocków, bloków). Połączenia określają kolejność i sposób wykonywania operacji realizujących dany algorytm. W literaturze informatycznej przyjęto pewne standardowe oznaczenia poszczególnych działań (są to figury geometryczne), ale można również używać innych oznaczeń (muszą one jednak być takie same dla określonego typu operacji).

Przykłady skrzynek do prezentacji algorytmu w postaci graficznej

Elementy schematu blokowego

Symbol graficzny	Nazwa skrzynki (bloku)	Funkcja	Opis
	Skrzynka graniczna	Początek algorytmu lub koniec	mają kształt owalu. Ze skrzynki START wychodzi tylko jedno połączenie, skrzynka STOP nie ma połączenia wychodzącego.
			
	Skrzynka operacyjna	Wykonywanie różnych działań, np. sumowania	ma kształt prostokąta.
	Skrzynka wejścia / wyjścia	Wprowadzanie (czytanie) danych lub wyprowadzanie (drukowanie, pisanie) wyników	jest równoległobokiem, wchodzi i wychodzi z niej jedno połączenie.
			
	Skrzynka warunkowa	Sprawdzanie warunku, np. czy $N > 0$	mają kształt rombu. Ze skrzynki wychodzą tylko dwa połączenia: jedno oznaczone TAK, a drugie NIE.

Zasady budowania schematu blokowego:

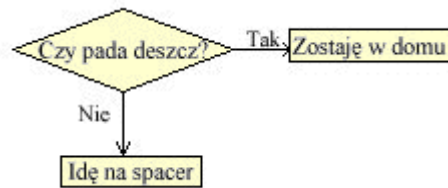
- Każda operacja jest umieszczona w skrzynce
- Schemat ma tylko jedną skrzynkę "Start" i przynajmniej jedną skrzynkę "Stop"
- Skrzynki są ze sobą połączone
- Ze skrzynki wychodzi jedno połączenie; wyjątek stanowią skrzynki "Stop" (z której nie wychodzą już żadne połączenia) oraz "warunkowa" (z której wychodzą dwa połączenia opisane Tak i Nie - w zależności od tego czy warunek jest spełniony czy też nie; można wyjść jedną z dwóch dróg).

Jak realizować sytuacje warunkowe ?

Przypomnij: co to jest instrukcja warunkowa ?

Z sytuacjami warunkowymi stykamy się w każdej dziedzinie życia codziennego. Na pytanie "Czy pada deszcz ?" odpowiedź może brzmieć "tak" lub "nie". W zależności od tego, czy warunek jest spełniony, czy nie, wybieramy inne rozwiązanie.

Przykład sytuacji warunkowej:



Ćwiczenie

1

Podaj przykłady sytuacji warunkowych. Przedstaw je graficznie.

Wiesz już, że z sytuacją warunkową mamy do czynienia wówczas, gdy wynik lub dalsze działanie zależy od spełnienia warunku. Na schemacie blokowym sytuacje warunkowe realizujemy przez skrzynkę warunkową. Jak dotąd rozważane przez nas algorytmy przedstawiały problemy z życia. Teraz zajmiemy się rozwiązywaniem prostych zadań obliczeniowych, przy czym algorytm opracowany dla każdego z nich będzie miał postać schematu blokowego.

W skrzynce wpisujemy warunek logiczny, stosując znaki "=" równy, "<>" różny, "<" mniejszy, ">" większy, "<=" mniejszy lub równy, ">=" większy lub równy, np: $(a > 5)$ lub $(a \leq 20)$, $(a < 5)$ OR $(a \leq 20)$
 Z systemami warunkowymi spotykamy się m.in. w matematyce i fizyce, wtedy gdy wykonywanie działań jest uzależnione od warunku, jakie muszą spełniać liczby, np. mają być nieujemne albo parzyste.

Ćwiczenie

2

Przeznacz schemat blokowy algorytmu obliczania wartości bezwzględnej.

Wartością bezwzględną liczby nieujemnej jest ta sama liczba, a wartością bezwzględną liczby ujemnej jest liczba do niej przeciwna.

Ćwiczenie

3

Wprowadź dwie liczby a i b . Przedstaw w postaci ciągu kroków i graficznie algorytm porównania tych liczb i wyprowadzania na wyjściu większej z nich. W przypadku liczb równych wyprowadź napis "liczby równe".

Na czym polega iteracja, czyli działanie w pętli ?

Przypomnij : co to są instrukcje iteracyjne ?
 Czasami trzeba wykonać te same operacje na wielu liczbach. W takich przypadkach nie jest konieczne wielokrotne opisywanie działań lub rysowanie takich samych skrzynek. Stosujemy wówczas iterację. Mówimy także, że działania te wykonywane są w pętli. Liczba powtórzeń tych działań może być z góry określona lub zależeć od spełnienia warunku. Iteracja to najczęściej spotykana technika algorytmiczna.

Iteracja to technika algorytmiczna polegająca na wykonaniu tej samej instrukcji dla n zmiennych.

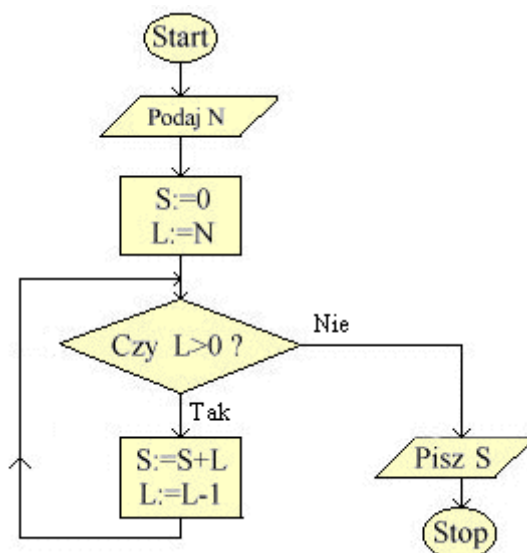
Iteracja oszczędza czas programisty, który ten musiałby spędzić wpisując instrukcję n razy, zależnie od liczby zmiennych. Liczba powtórzeń w iteracji jest zwykle z góry określona lub zależy od spełnienia określonego warunku.

Ćwiczenie 4

Przeanalizuj schemat i odpowiedz na pytania:

1. *Jakie są dane wejściowe do zadania i jakich użyto zmiennych pomocniczych?*
2. *Jaki cel chcesz osiągnąć (wynik)?*
3. *Jaki algorytm zastosowano w operacji dodawania liczb ?*
4. *W którym miejscu wykonywane są działania w pętli ?*
5. *Które operacje powtarzają się wielokrotnie ?*
6. *Co określa liczbę powtórzeń ?*
7. *Kiedy kończy się działanie w pętli ?*

W analizowanym algorytmie występuje przypisanie typu $s:=s + l$. Nie jest to równość w rozumieniu matematyki, ale przypisanie zmiennej "s" (sumie) poprzedniej wartości pamiętanej w zmiennej "s" (poprzedniej sumie) zwiększonej o wartość pamiętaną w zmiennej "l" (kolejną liczbę) - w ten sposób powtarzanie operacji przypisania, realizowane jest dodawanie kolejnych liczb naturalnych. Przypisanie $l:=l-1$ ma tu bardzo istotne znaczenie. Jest to tzw. licznik, w którym następuje obliczanie, ile zostało jeszcze powtórzeń do wykonania. W tym algorytmie liczba powtórzeń została określona na początku instrukcją $l:=n$.



Gdyby nie umieszczono tego działania nastąpiło by tzw. zapętlenie algorytmu, czyli po sprawdzeniu warunku $l>0$ działanie w schemacie przebiegałoby zawsze drogą TAK. W realizacji algorytmów iteracyjnych ważne jest prawidłowe określenie sposobu

zakończenia działań. Można to zrobić za pomocą licznika, który odlicza kolejne kroki iteracji (liczbę powtórzeń).

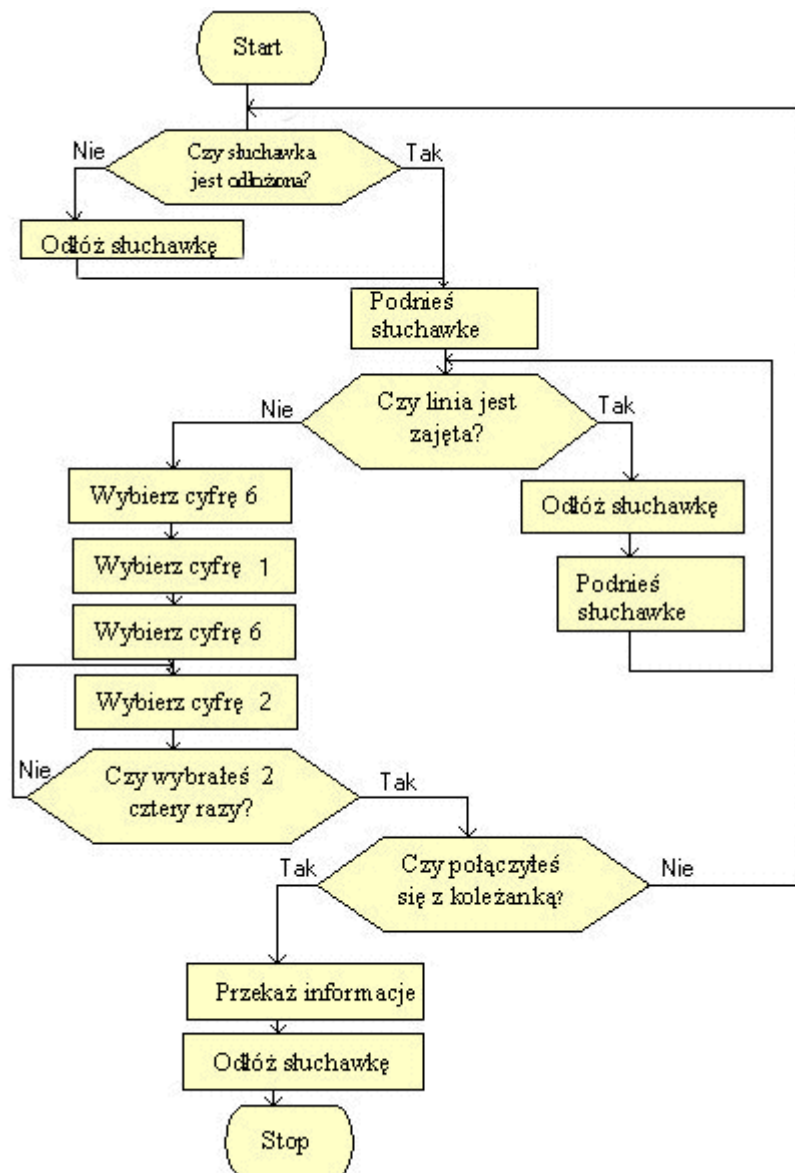
Ćwiczenie **5**
Narysuj schemat blokowy algorytmu telefonicznego zaproszenia koleżanki w wersji pierwszej

Ćwiczenie **6**
Narysuj w postaci schematu blokowego algorytm telefonicznego zaproszenia koleżanki w wersji drugiej

Ćwiczenie **7**
Narysuj w postaci schematu blokowego algorytm telefonicznego zaproszenia koleżanki w wersji trzeciej .

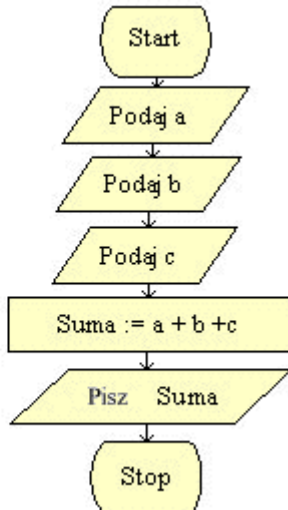
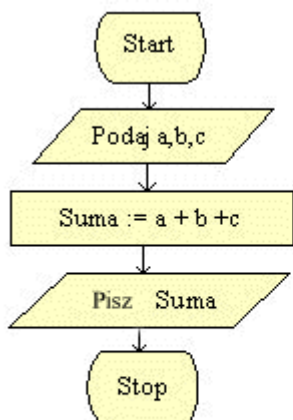
Ćwiczenie **8**
Narysuj w postaci schematu blokowego algorytm telefonicznego zaproszenia koleżanki w wersji czwartej.

Ćwiczenie **9**
Narysuj w postaci schematu blokowego algorytm telefonicznego zaproszenia koleżanki w wersji piątej.



Podsumujmy:

1. **Zapis algorytmu w postaci ciągów kroków.** Sposób rozwiązania zadania, czyli algorytm, można podać w kilku punktach. Mówimy wówczas, że algorytm jest opisany za pomocą ciągu kroków. Zapis taki polega na podaniu kolejnych wykonywanych operacji, składających się na całe rozwiązanie.
2. **Zapis algorytmu w postaci graficznej - schematy blokowe.** Schemat blokowy to graficzne przedstawienie ciągu kroków algorytmu, często stosowane jako ideowy rysunek poprzedzający tworzenie programu. Sposób i kolejność działań programu określone są przez wzajemny układ i sposób łączenia bloków ze sobą. Każde działanie (krok) ma w schemacie blokowym swoje standardowe oznaczenie (patrz tabela wyżej).



Algorytmy - przykłady

Zachęcam Cię, abyś po przeczytaniu treści ćwiczenia zastanowił się nad rozwiązaniem, a następnie zbudował schemat. Dopiero po samodzielnej pracy porównaj i dokonaj analizy

gotowego schematu przedstawionego na stronie.

Ćwiczenie

1

Opracuj algorytm obliczający sumę 3 wprowadzonych z klawiatury liczb.

Przedstawmy najpierw algorytm w postaci ciągu kroków do wykonania:

1. Podaj pierwszą liczbę
2. Podaj drugą liczbę
3. Podaj trzecią liczbę
4. Dodaj do siebie liczby i wynik zapamiętaj
5. Wypisz otrzymany wynik

lub

Czy zwróciłeś uwagę na operację $Suma := a+b+c$?
Znak przypisania $:=$ oznacza nadanie zmiennej znajdującej się po jego lewej stronie wartości określonej po prawej stronie.

Ćwiczenie

2

Dane są dwie liczby. Opracuj algorytm wybierający większą z nich, w przypadku gdy są równe wypisz stosowny komunikat.

Ćwiczenie

3

Zbuduj algorytm obliczający sumę 5 kolejnych liczb naturalnych.

Ćwiczenie

4

Wyznacz NWD- największy wspólny dzielnik dwóch liczb naturalnych a, b .

Wyznaczanie największego wspólnego dzielnika dwóch liczb naturalnych zrealizujemy za pomocą algorytmu Euklidesa. Opiera się on na spostrzeżeniu, że jeśli odejmiemy od większej liczby mniejszą, to ta mniejsza liczba i otrzymana różnica będą miały taki sam największy wspólny dzielnik jak pierwotne liczby. Jeśli w wyniku kolejnego odejmowania otrzymamy parę równych liczb, oznacza to, że znalazłeś NWD.

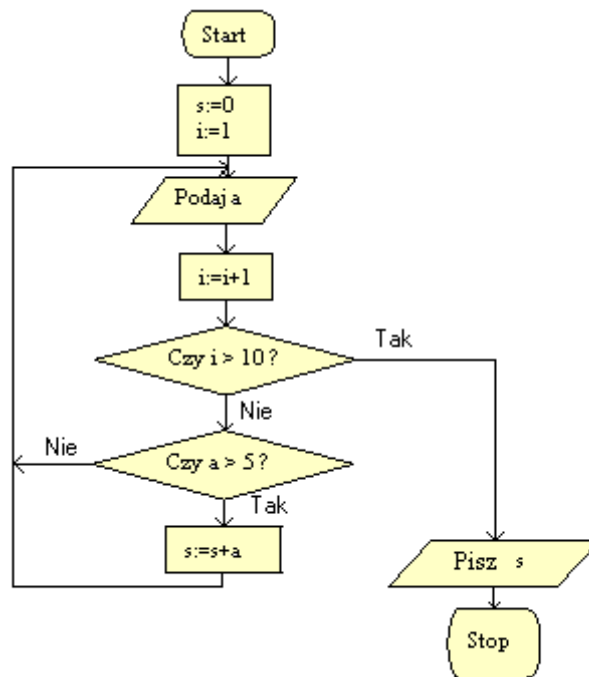
Przykład

Znajdź największy wspólny dzielnik liczb 20 i 50.
 $a=20$ $b=50$
 $a < b$
 a (b
 $b = b - a = 50 - 30 = 20$
 $a = 20$ $b = 10$
 $a < b$
 $a > b$
 $a = a - b = 20 - 10 = 10$
 $a = 10$ $b = 10$
 $a = b$
 $NWD(50, 20) = 10$

Ćwiczenie

5

Narysuj algorytm sumujący liczby większe od 5 spośród 10 wprowadzonych.



Zrób sam:

Ćwiczenie **1**
Opracuj algorytm obliczający kwadrat i sześcian wprowadzonej liczby.

Ćwiczenie **2**
Opracuj schemat obliczania średniej n liczb naturalnych.

Algorytmy sortowania

W życiu codziennym często spotykamy się z faktem porządkowania. Na pewno korzystałeś już ze słowników, czy encyklopedii. Aby łatwiej było znaleźć interesujące nas hasło, są one uporządkowane według określonej zasady.

Sortowanie - proces ustawienia zbioru obiektów w określonym porządku

Metody sortowania:

- przez zamianę (sortowanie bąbelkowe)
- przez wybieranie
- przez wstawianie

Sortowanie bąbelkowe

Ćwiczenie 1
Uporządkuj cztery dowolne liczby od najmniejszej do największej.
Jedną z popularnych metod sortowania jest sortowanie przez prostą zamianę (sortowanie bąbelkowe). W metodzie tej porównujemy sąsiednie elementy. W celu uporządkowania elementów od najmniejszego do największego, jeśli drugi element jest mniejszy od poprzedniego, to zamieniamy go miejscami. Następnie element, który stał się drugim, porównujemy z trzecim i przestawiamy, jeśli jest mniejszy itd.

Przykład:

Przebieg pierwszy:

```
7 6 1 9 5 - porównujemy 7 i 6 ( przestawiamy, bo 6 < 7)
6 7 1 9 5 - porównujemy 7 i 1 ( przestawiamy, bo 1 < 7)
6 1 7 9 5 - porównujemy 7 i 9 ( nie przestawiamy, bo 10 > 7)
6 1 7 9 5 - porównujemy 9 i 5 ( przestawiamy, bo 5 < 10)
6 1 7 5 9 - 9 na właściwym miejscu
```

Przebieg drugi:

```
6 1 7 5 9 - porównujemy 6 i 1 ( przestawiamy, bo 1 < 6)
1 6 7 5 9 - porównujemy 6 i 7 ( nie przestawiamy, bo 7 > 6)
1 6 7 5 9 - porównujemy 7 i 5 ( przestawiamy, bo 5 < 7)
1 6 7 5 9 - porównujemy 7 i 5 ( przestawiamy, bo 5 < 7)
1 6 5 7 9 - 7 i 9 na właściwym miejscu
```

Przebieg trzeci:

```
1 6 5 7 9 - porównujemy 1 i 6 ( nie przestawiamy, bo 6 > 1)
1 6 5 7 9 - porównujemy 6 i 5 ( przestawiamy, bo 5 < 6)
1 5 6 7 9 - 6, 7, 9 na właściwym miejscu
```

Przebieg czwarty:

```
1 5 6 7 9 - porównujemy 1 i 5 nie przestawiamy, bo 5 > 1
...
```

w czwartym przebiegu nie są zamieniane żadne elementy, czyli nie ma potrzeby dalszego porównywania. Można zakończyć algorytm, liczby są uporządkowane w żądanej kolejności.

Spójrz na rysunek. Czy ciąg po pierwszym przebiegu jest już uporządkowany? Ponieważ nie jest, należy wykonywać kolejne przebiegi. Powinniśmy to robić, aż liczby zostaną uporządkowane, a więc do momentu, aż nie trzeba będzie przestawiać miejscami żadnego elementu.

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

Ćwiczenie

2

Ułóż 5 książek w przypadkowej kolejności na stole. Uporządkuj je według wysokości (od najwyższej do najniższej), stosując algorytm sortowania bąbelkowego.

Sortowanie przez wybieranie

Jeśli chcemy uporządkować elementy jakiegoś zbioru np. w kolejności od najmniejszego do największego to możemy wybrać najmniejszy element i umieścić go na początku. Następnie wybieramy najmniejszy element z pozostałego zbioru i umieszczamy go po poprzednio wybranym elemencie (czyli jako drugi) itd. Taka metoda porządkowania nazywa się sortowaniem przez wybór. Polega na wyszukaniu największej liczby, przestawieniu jej na początek ciągu elementów (czyli zamienieniu jej z pierwszą liczbą ciągu) i takim samym postępowaniu dalszym z pominięciem pierwszego elementu.

```
7 6 1 9 5 - 1 na pierwszą pozycję, 7 w miejsce 1
1 6 7 9 5 - 5 na drugą pozycję, 6 w miejsce 5
1 5 7 9 6 - 6 na trzecią pozycję, 7 w miejsce 6
1 5 6 9 7 - 7 na czwartą pozycję, 9 w miejsce 7
1 5 6 7 9 - 9 na właściwym miejscu (piątym)
```

Ćwiczenie

3

Wykonaj zadanie z ćwiczenia drugiego, stosując algorytm sortowania przez wstawianie.

Sortowanie przez wstawianie

Metoda porządkowania przez wybór polega na wstawianiu elementu we właściwe miejsce, jest ona powszechnie stosowana przez osoby grające w karty.

```
7 | 6 1 9 5
6 7 | 1 9 5
1 6 7 | 9 5
1 6 7 9 | 5
1 6 7 5 9 |
```

Na schemacie kreska odziela posortowaną lewą część liczb od nieposortowanej prawej części.

Jeśli pragniemy ustawić liczby w kolejności od najmniejszego do największego, to bierzemy element drugi i jeśli jest on mniejszy od pierwszego to wstawiamy przed nim. Następnie bierzemy element trzeci i wstawiamy również w odpowiednie miejsce, a więc jeśli jest mniejszy od pierwszego to przed pierwszy, jeśli mniejszy od drugiego to przed drugi.

Ćwiczenie

3

Wylosuj sześć kart, następnie uporządkuj je w kolejności od największej do najmniejszej.

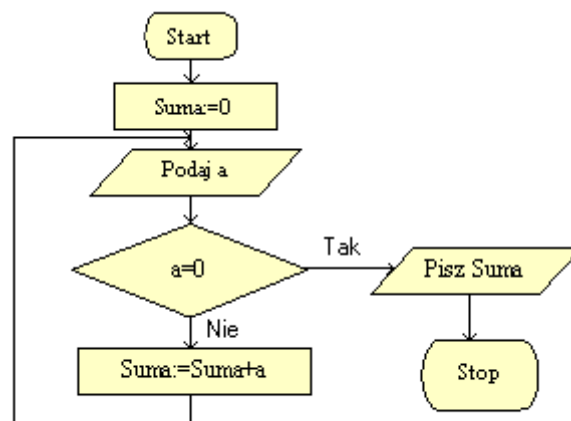
Sortowanie na przykładzie tablic

Ciągi w algorytmach

Ćwiczenie

1

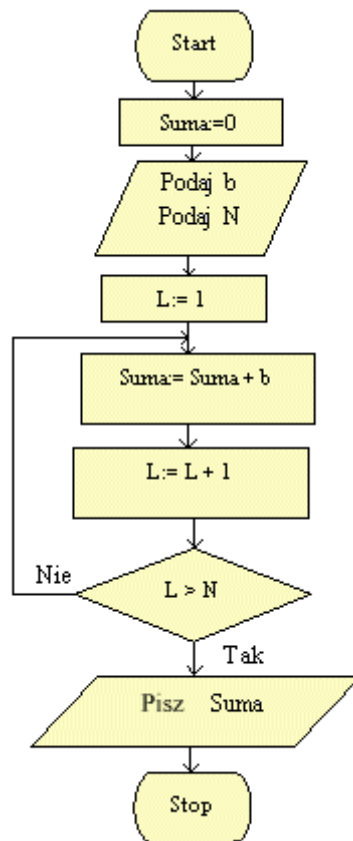
Dany jest ciąg liczb o nieznannej długości. Ostatnia liczba w ciągu równa się zero. Oblicz sumę elementów tego ciągu.



Ćwiczenie

2

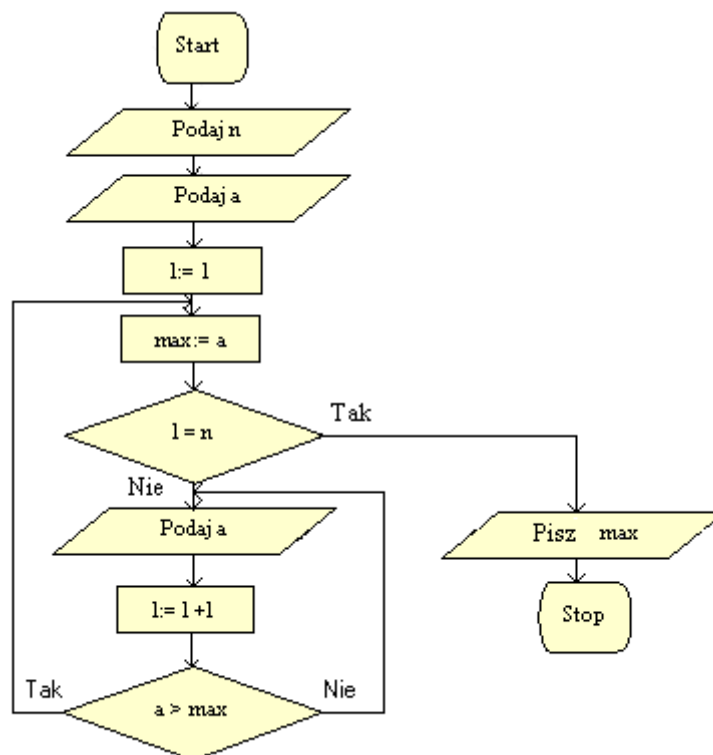
Oblicz sumę n - kolejnych liczb naturalnych. Pierwsza z liczb wprowadzona z klawiatury i równa się b .

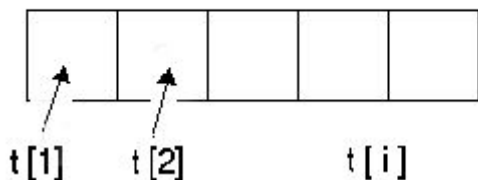


Ćwiczenie

3

Dany jest ciąg n -elementowy. Elementy tego ciągu są różne. Wskaż największy element ciągu.





Zrób sam:

Dany jest ciąg liczb o nieznannej długości. Ostatnia liczba w ciągu równa się zero. Oblicz średnią arytmetyczną elementów ciągu.

Algorytmy obsługi tablic

Tablicą nazywamy złożoną strukturę danych, która zawiera zbiór elementów tego samego typu.

Wyróżniamy:

- tablice jednowymiarowe
- tablice wielowymiarowe

Tablice jednowymiarowe

Tablica jest to zbiór elementów tego samego typu. Każdy element tablicy jest identyfikowany przez jego numer (indeks). Każdy element tablicy posiada swoją wartość.

Oto tablica t zawierająca 5 elementów:

5	11	8	3	2
1	2	3	4	5

Do poszczególnych elementów tablicy uzyskujemy dostęp poprzez podanie nazwy tablicy oraz w nawiasach kwadratowych wartość indeksu (numer żadanego elementu).

I	tak	w	naszej	tablicy	o	nazwie	t :
		t		[1]	=		5
		t		[2]			=11
		t		[3]			=8
		t		[4]			=3
		t		[5]			=2

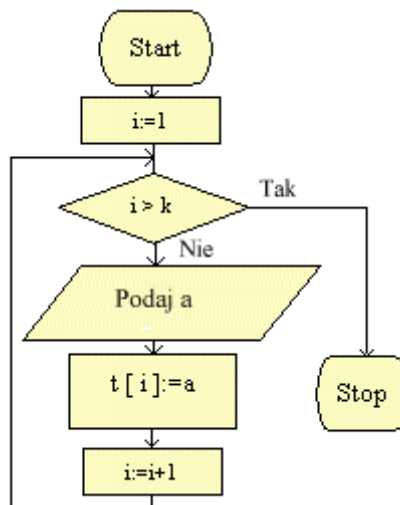
Element $t[i]$ dla i równego 4 wynosi 3, dla i równego 2 wynosi 11.

Ćwiczenie

1

Opracuj algorytm wypełniający tablicę k - elementową.

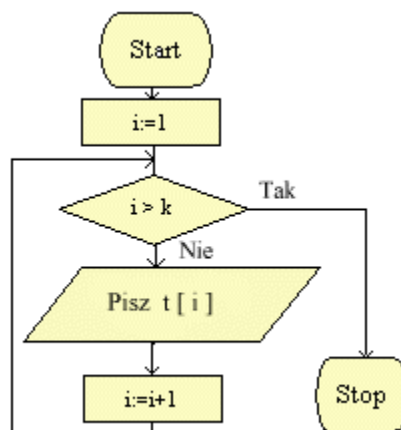
Ponieważ tablica ma k - elementów, a więc k - razy nastąpi wczytywanie elementu, stąd zastosowanie iteracji.



Ćwiczenie

Opracuj algorytm wypisujący elementy tablicy k - elementowej.

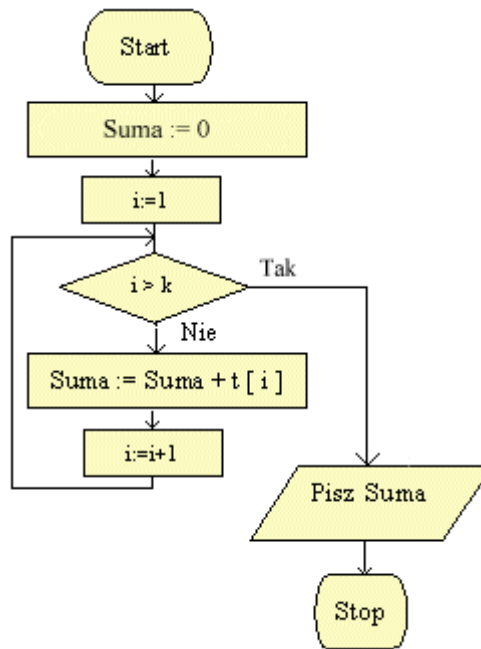
2



Ćwiczenie

Zbuduj algorytm sumujący elementy tablicy n - elementowej.

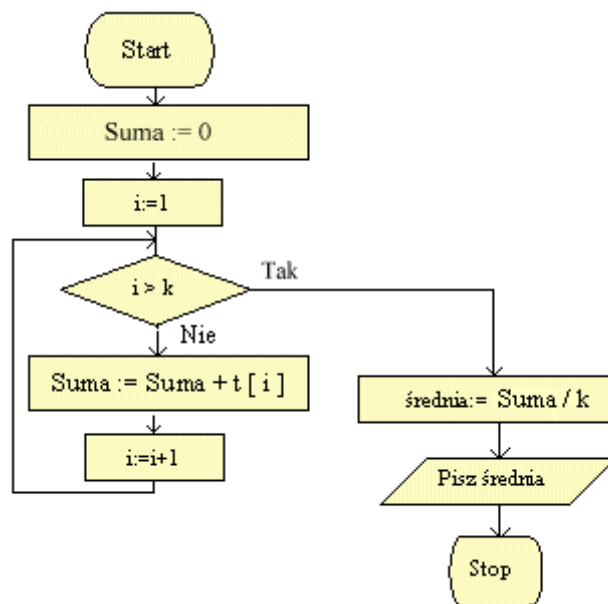
3



Ćwiczenie

Zbuduj algorytm liczący średnią arytmetyczną elementów tablicy.

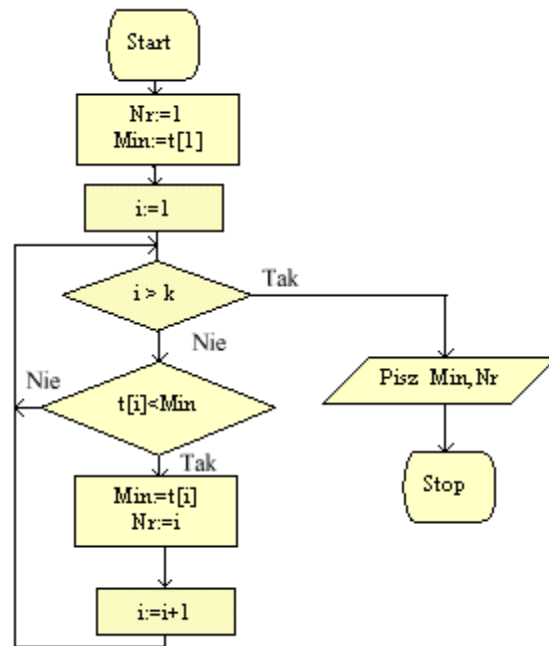
4



Ćwiczenie

Do tablicy wczytano k-elementów. Wydrukuj położenie i wartość elementu najmniejszego (zakładamy, że elementy są różne).

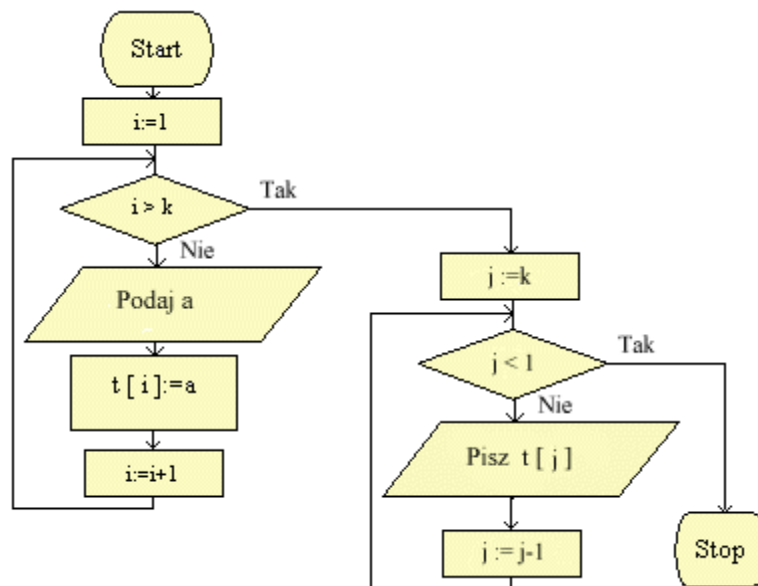
5



Ćwiczenie

6

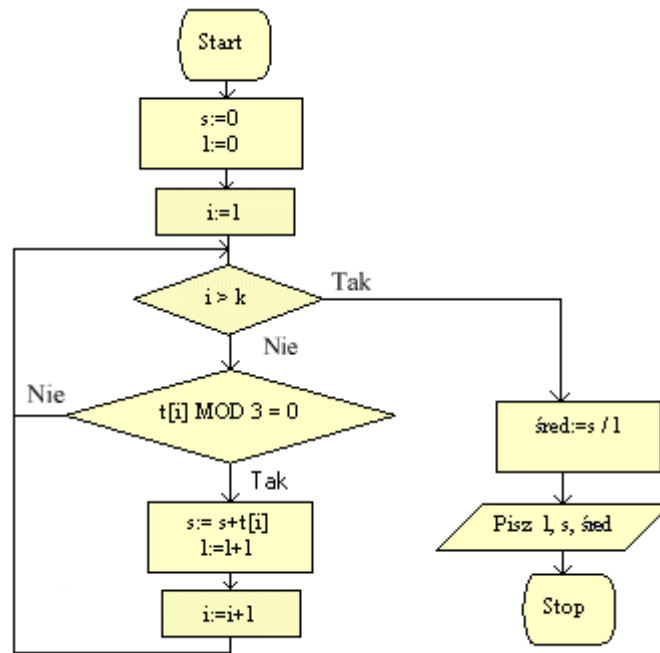
Wczytaj elementy do tablicy k -elementowej, a następnie wydrukuj je w odwrotnej kolejności.



Ćwiczenie

7

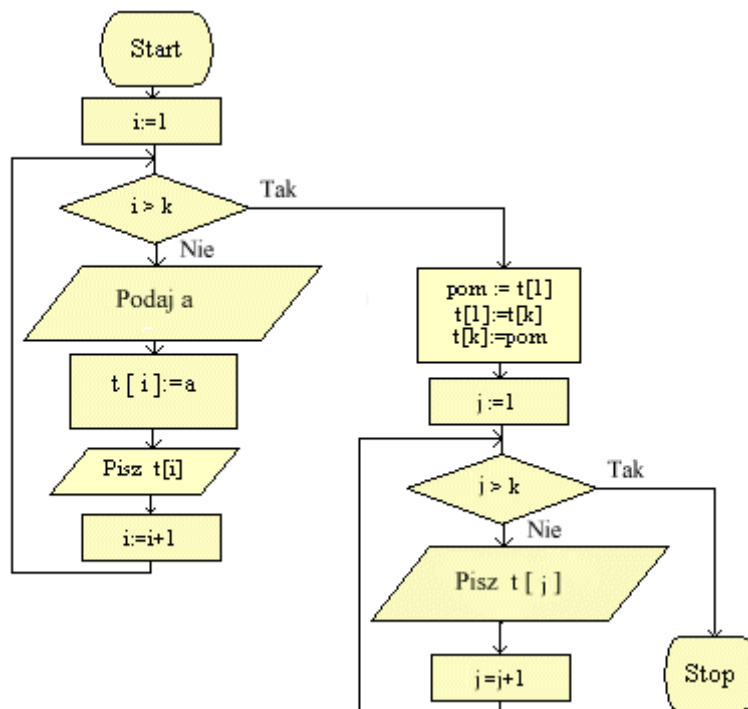
Z tablicy zawierającej k -elementów wybierz elementy podzielne przez 3. Oblicz ich ilość, sumę i średnią.



Ćwiczenie

8

Wczytaj elementy do tablicy k- elementowej. Wydrukuj tę tablicę. Zamień pierwszy z ostatnim i wydrukuj tablicę po modyfikacji.

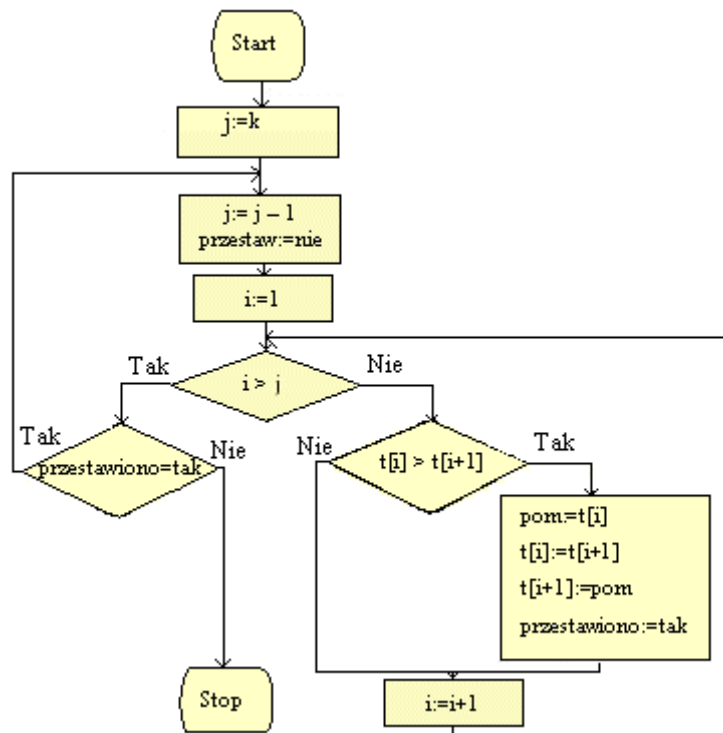


Ćwiczenie

9

Uporządkuj elementy tablicy k- elementowej, w kolejności od najmniejszego do największego.

Aby uporządkować tablicę posłużymy się jedną z metod sortowania, na przykład sortowaniem bąbelkowym.



Rozwi \acute{a} ż samodzielnie:

Ćwiczenie 1
Opracowany w ćwiczeniu 3 algorytm zmodyfikuj tak, by wynikiem była suma elementów większych od 10.

Ćwiczenie 2
Opracuj algorytm liczący średnią arytmetyczną elementów dodatnich tablicy.

Ćwiczenie 3
Zbuduj algorytm wyszukiwania elementu maksymalnego w tablicy.

Ćwiczenie 4
Opracuj algorytm

1. wczytujący elementy tablicy k-elementowej
2. wyszukujący element minimalny
3. wypisujący elementy tablicy i element minimalny

Algorytmy obsługi tablic dwuwymiarowych

Przykład tablicy dwuwymiarowej:

	1	2	3	4	
1	2	6	11	0	wiersze
2	7	5	9	-6	
3	9	1	2	3	
					kolumny

Tablica ma 3 wiersze i 4 kolumny

Do poszczególnych elementów tablicy uzyskujemy dostęp poprzez podanie nazwy tablicy oraz w nawiasach kwadratowych wartość indeksu (numer żądanego elementu). I tak w naszej tablicy o nazwie t zapis t[3,4] oznacza element w trzecim wierszu i czwartej kolumnie, czyli element o wartości =3.

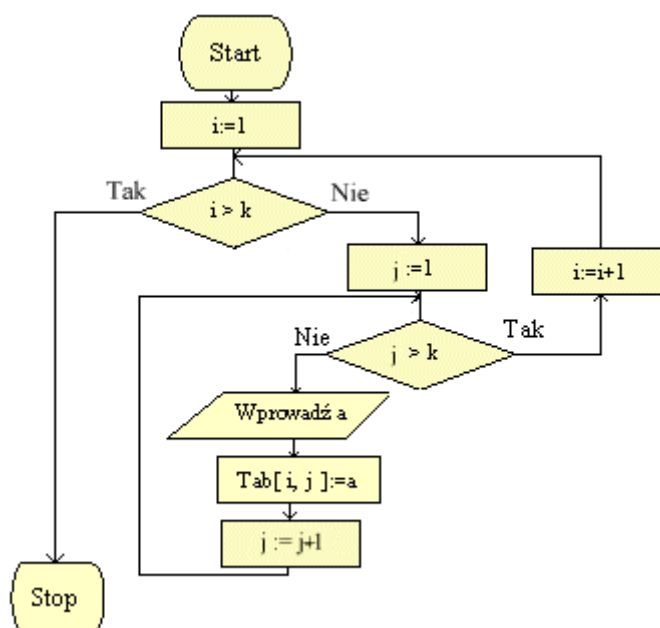
t [1,1] = 2
t [1,2] = 6
t [1,3] = 11
t [1,4] = 0
t [2,1] = 7

Element t [i, j] dla i równego 2 i j równego 2 wynosi 5.

Ćwiczenie

1

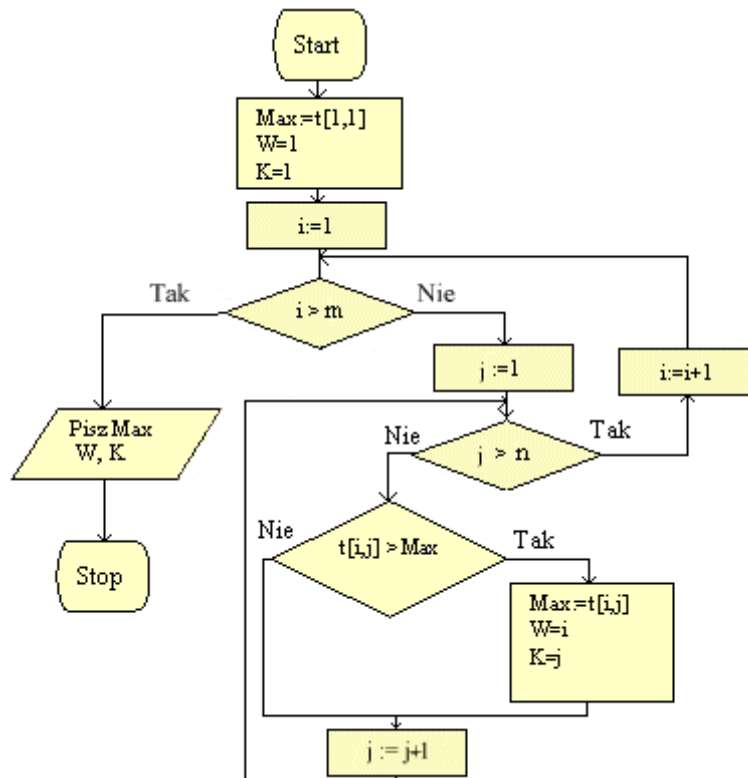
Opracuj algorytm wypełniający tablicę kwadratową k x k.



Ćwiczenie

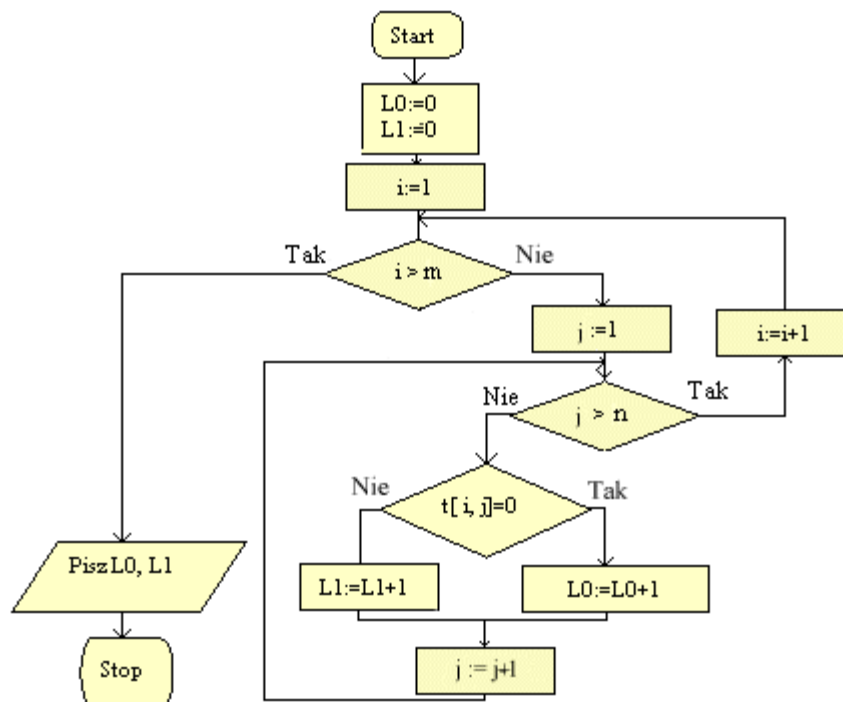
2

Odszukaj w tablicy $m \times n$ element maksymalny. Wypisz jego położenie i wartość.

**Ćwiczenie**

3

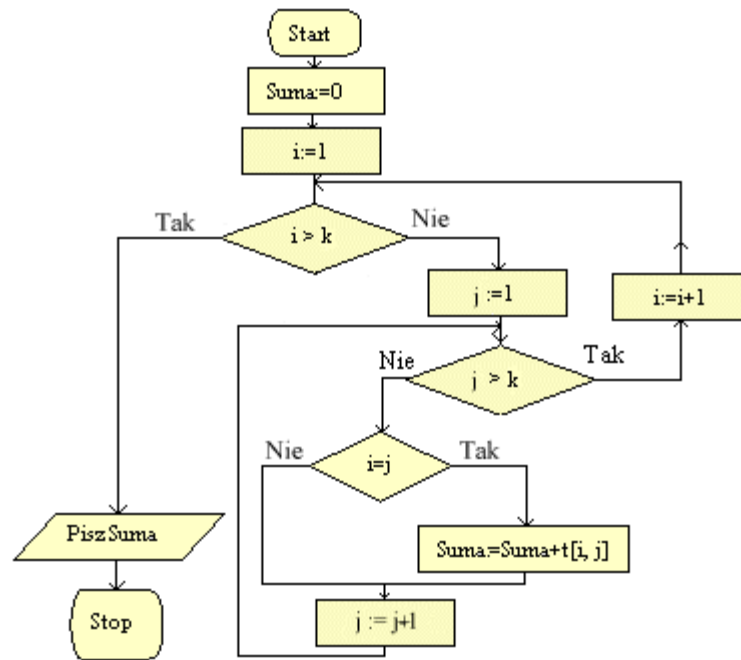
Tablica kwadratowa $m \times n$ składa się z zer i jedynek. Podaj algorytm wypisujący ile jest zer i ile jedynek.



Ćwiczenie

4

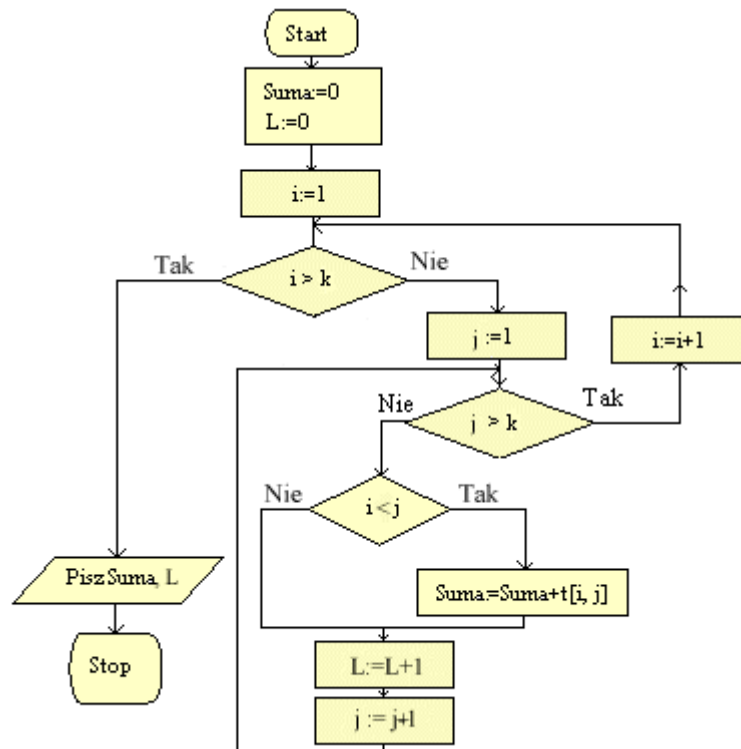
Zbuduj algorytm sumujący elementy na przekątnej tablicy $k \times k$.



Ćwiczenie

5

Podaj algorytm obliczający sumę i ilość elementów nad przekątną tablicy kwadratowej.



Rozwiąż samodzielnie:

Ćwiczenie **1**
Zbuduj algorytm wyznaczania elementu minimalnego tablicy kwadratowej.

Ćwiczenie **2**
Narysuj schemat blokowy algorytmu wypisującego z tablicy $k \times k$ elementy parzyste. Podaj ilość takich elementów.

Ćwiczenie **3**
Zbuduj algorytm podający ilość i średnią arytmetyczną elementów tablicy $m \times n$ podzielnych przez 10.

Ćwiczenie **4**
Opracuj algorytm wyznaczania elementu minimalnego tablicy kwadratowej.