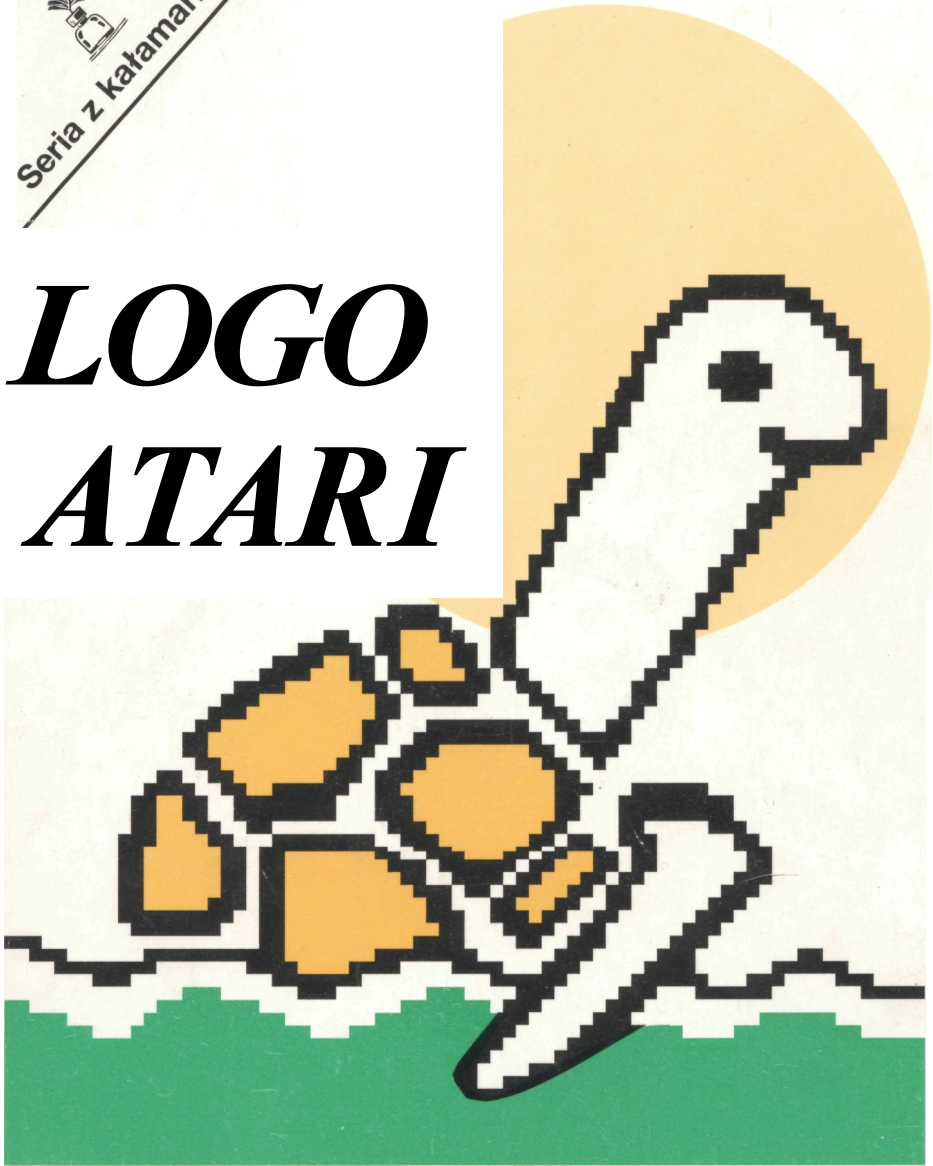




Seria z kałamarzem

Bronisław FETRAŚ

LOGO ATARI

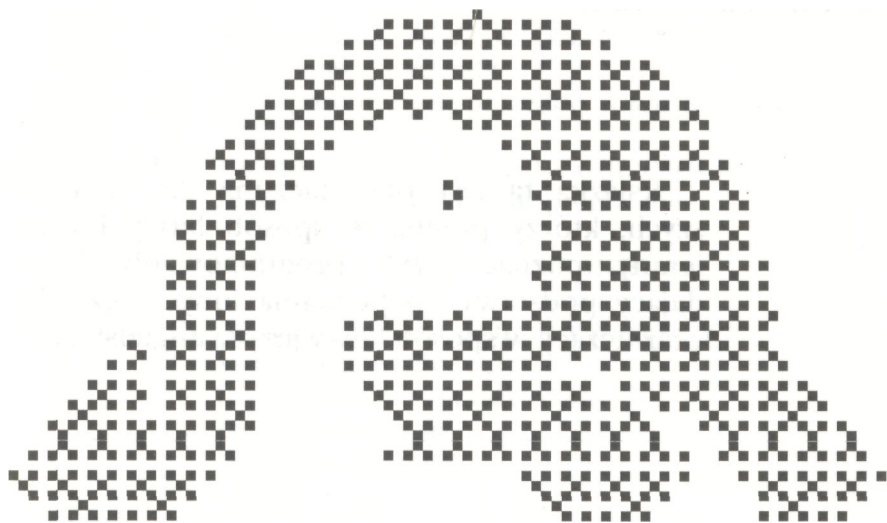


czyli

Wprowadzenie do programowania

Bronisław FETRAŚ

LOGO ATARI



czyli

*Wprowadzenie
do Programowania*

Redaktor wydania: Mieczysław JAKUBICKI

Okładkę projektowała: © Małgorzata JAKUBICKA

Skład komputerowy wykonano w:



Agencja Wydawnicza
o2-792 W-wa, ul. Lanciego 12
tel. 643-25-09
program DMC Calamus

Autor: Bronisław FETRAŚ

Książka ta jest przeznaczona dla najmłodszych, którzy pragną w sposób łatwy i przyjemny poznać język programowania Logo, dający podstawy do poznania innych języków oraz uczący w sposób łatwy języka angielskiego.

ISBN 83-85424-05-9

© Copyright M&M Agencja Wydawnicza
Warszawa 1991

SPIS TREŚCI

ROZDZIAŁ I:	"JAK ZACZYNAĆ" Wczytywanie ATARI LOGO Klawiatura	5
ROZDZIAŁ II:	"KOMENDA PRINT" Czyszczenie ekranu Tworzenie procedur	9
ROZDZIAŁ III:	"SPOTKANIE Z ŻÓŁWIEM" Zmiana stanu żółwia	14
ROZDZIAŁ IV:	"JAK NAUCZYĆ ŻÓŁWIA RYSOWAĆ KWADRAT" Wprowadzenie do ATARI LOGO EDITOR Użycie nowej komendy Inne sposoby użycia procedury KWADRAT	19
ROZDZIAŁ V:	"ZAPISYWANIE I ODTWARZANIE PROGRAMÓW" Zapisywanie i odtwarzanie programów	27
ROZDZIAŁ VI:	"RYSUNKI I TEKST NA EKRANIE"	29
ROZDZIAŁ VII:	"PIÓRO ŻÓŁWIA I KOLOR RYSUNKÓW" Komendy dotyczące pióra Użycie kolorów w grafice ATARI Zmiana kolorów tła Zmiana koloru pióra	30
ROZDZIAŁ VIII:	"NOWE INFORMACJE O UŻYCIU EDYTORA" Wyjście z Edytora Redagowanie poza Edytorem	36
ROZDZIAŁ IX:	"PRZESTRZEŃ ROBOCZA" Odczytywanie procedur Usuwanie procedur z przestrzeni roboczej	39
ROZDZIAŁ X:	"PROJEKT PROCEDURY RYSUJĄCEJ PAJĄKA"	42

ROZDZIAŁ XI:	"ELEMENTY ŻÓŁWIOWEJ GEOMETRII TRÓJKĄTY" Namiot w drzewie Żółw buduje dom	46
ROZDZIAŁ XII:	"ZMIENNE. MAŁE I DUŻE KWADRATY" Kilka przykładów wykorzystania PUDELKOP Duże i małe trójkąty Arytmetyka Liczby w LOGO	52
ROZDZIAŁ XIII:	"ŁUKI I OKRĘGI" Okrag o żądanym promieniu Żółw rysuje łuki	60
ROZDZIAŁ XIV:	"JAK WYGLĄDA POLE ŻÓŁWIA" WRAP I WINDOW Użycie operacji POS w rysowaniu	65
ROZDZIAŁ XV:	"W KRAINIE WIELOKĄTÓW I SPIRALI" Spirale	70
ROZDZIAŁ XVI:	"SPECJALNE MOŻLIWOŚCI GRAFIKI ŻÓŁWIA W ATARI LOGO" Cztery żółwie Żółwie w ruchu Kolor żółwia Kształt żółwia Wykrywanie zderzeń	73
ROZDZIAŁ XVII:	"PROJEKTOWANIE GIER" Struktura gry Przystosowanie klawiszy do sterowania grą Rozszerzenie projektu gry	81
ROZDZIAŁ XVIII:	"PROCEDURY REKURENCYJNE" Zatrzymanie procedur rekurencyjnych	88
DODATEK A		91
DODATEK B		92
DODATEK C		93

ROZDZIAŁ I

"JAK ZACZYNAĆ"

Język programowania LOGO w porównaniu z takimi językami jak francuski lub angielski posiada niewiele słów i prawideł gramatycznych. LOGO działa opierając się na pewnej ilości tzw. słów kluczowych, prostych komend wbudowanych w język. Pozwalają one wykonywać operacje dodawania, odejmowania, mnożenia i dzielenia oraz dają możliwość manipulacji słowami i literami.

ATARI LOGO jest językiem łatwym w użyciu, słowa kluczowe i pierwsze stworzone (przez Ciebie) procedury pozwolą Ci zbudować inne, bardziej złożone programy.

Ten podręcznik zawiera niezbędne wiadomości potrzebne początkującym, chociaż nie jest pełnym opisem ATARI LOGO. Informacje zaprezentowane tutaj dadzą Ci pojęcie o możliwościach języka wyższego rzędu. Szybko nauczysz się pisać procedury, używać Edytora, zapisywać i odtwarzać swoje programy.

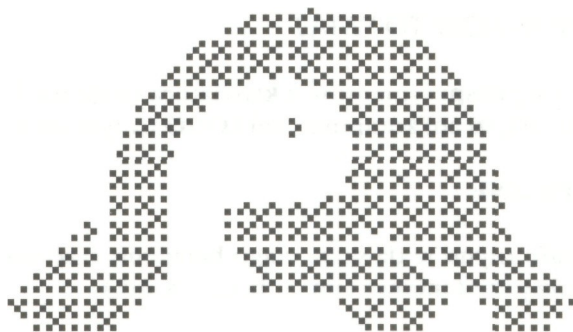
Podręcznik, który trzymasz w ręku opowiada o Grafice Żółwia. Stanowi ona przyjemne wprowadzenie do programowania.

W rozdziałach 1-4 znajdziesz podstawy uruchomienia systemu i wykonasz pierwsze kroki w poznaniu programu.

W rozdziałach 5-15 *zaczyniesz* poruszać się w magicznym świecie geometrii żółwia.

W rozdziałach 16 i 17 oraz dodatkach B i C będziemy rozważać te aspekty Grafiki Żółwia, które są specyficzne dla ATARI LOGO.

Tyle tytułem wstępu - a teraz zapraszamy Ciebie na spotkanie z:



WCZYTYWANIE ATARI LOGO

Chcąc wczytać interpreter języka ATARI LOGO do pamięci swojego komputera musisz postępować zgodnie z zaleceniami dystrybutora języka LOGO. Jest to zależne od tego czy kupisz LOGO ATARI na kasecie, dysku lub cartridgu.

Po wczytaniu Logo do komputera na ekranie swojego telewizora lub monitora ujrzysz napis:

```
(C) 1983 LCSII ALL RIGHTS RESERVED  
WELCOME TO ATARI LOGO  
? ■
```

Oznacza to koniec wczytywania i początek Twojej przygody. Znak "?" jest znakiem zachęty. Gdy widzisz go na ekranie oznacza to, że komputer oczekuje na Twoje polecenia. Mały jasny prostokąt na ciemnym tle ekranu, nazywamy kursorem, pokazuje miejsce w którym pojawi się następny znak wprowadzany przez Ciebie z klawiatury.

KLAWIATURA

Klawiatura komputera ATARI przypomina klawiaturę maszyny do pisania. Dla wprawy napisz cokolwiek na ekranie - niekoniecznie coś mającego w LOGO sens.

WITAJ - Naciśnij RETURN

LOGO odpowie:

```
I DONT KNOW HOW TO WITAJ
```

Nie obawiaj się eksperymentów z klawiaturą (o ile nie będziesz używał klawisza RESET, to ani komputer, ani LOGO na tym nie ucierpią).

ZNAKI GRAFICZNE

Znakami graficznymi w dalszym ciągu będziemy nazywać litery alfabety, cyfry, znaki przestankowe (np. A, B, C, 7, \$, itd.)



RETURN

Naciśnięcie tego klawisza mówi LOGO: "Wykonaj to, co napisałem", służy do wprowadzania instrukcji.

SPACJA

Wciśnięcie klawisza spacji powoduje wpisanie na ekranie niewidocznego, ale bardzo ważnego znaku spacji. W LOGO spacja służy do oddzielania poszczególnych komend i słów. Na przykład LOGO rozumie "DZIENDOBRY" napisane razem jako jedno słowo, a "DZIEN DOBRY" jako dwa słowa.

SHIFT

Wciśnięcie SHIFT jednocześnie z którymś ze znaków graficznych zmienia ich znaczenie np. po naciśnięciu SHIFT LOGO wypisze na ekranie "[" — w przeciwnym razie pojawi się przecinek.

Uwaga!

Znaki kwadratowych nawiasów "[" , "]" są w LOGO bardzo ważne i nie należy ich mylić z nawiasami okrągłymi "(, ")".

CTRL (CONTROL)

Pozwala on zmienić klawisze znaków graficznych w klawisze funkcji, naciśnięcie samego CTRL nic nie wywołuje. Użycie takiej kombinacji nie zawsze pozostawia ślad na ekranie, ale zawsze jest rozumiane przez komputer.



CTRL + STRZAŁKA

CTRL← porusza kursor o jedną pozycję w lewo,

CTRL→ porusza kursor o jedną pozycję w prawo

Są one pomocne przy redagowaniu tekstu programu. Kursor porusza się nie naruszając napisów.

CTRL↑, CTRL↓ działają analogicznie, ale tylko w Edytorze LOGO. Umieszczając kursor w odpowiednim miejscu możemy dokonywać zmian oraz kasowania znaków:

aby zmienić znak umieszczony kursor na następnej pozycji za znakiem in naciskamy DELETE BACK S. Następnie wpisujemy nowy znak (który zajmuje miejsce usuniętego).

BREAK

Powoduje przerwanie aktualnie wykonywanych przez LOGO czynności. W szczególności powoduje wyjście z Edytora bez uwzględnienia dokonanych zmian. Po naciśnięciu "BREAK" na ekranie pojawi się.

STOPPED!

?



a komputer oczekuje na następne instrukcje.

ESC

Jest on także używany do wyjścia z Edytora. Szersze omówienie działania tego klawisza znajduje się w rozdziale 4.

Klawisz 

Naciśnięcie tego klawisza powoduje wyświetlenie na ekranie negatywów wszystkich następnych napisanych znaków (tzn. ciemny znak na jasnym tle.). Do zwykłego sposobu pisania można powrócić poprzez ponowne naciśnięcie powyższego klawisza.

CAPS

Kiedy pierwszy raz włączyłeś swój ATARI, cokolwiek napisałeś pojawiło się na ekranie dużymi literami. Po naciśnięciu "CAPS" pojawiają się wyłącznie małe litery. Zatem jeśli przypadkowo naciśniesz ten klawisz, LOGO przestanie rozumieć Twoje instrukcje.

RESET



Nie używaj tego klawisza! Zniszczysz całą zawartość pamięci.

"KOMENDA PRINT"

Zapoznanie się z LOGO rozpoczniemy od komendy PRINT. Żeby wypróbować jej działanie napisz:

```
PRINT [JAK SIE MASZ?]
```

Instrukcja pojawia się na ekranie, ale nie jest wykonywana, aż do momentu gdy wciśniesz klawisz RETURN. Wtedy LOGO odpowie:

```
JAK SIE MASZ
```

ale co zrobić w razie pomyłki?

Założmy, że chciałeś, żeby LOGO napisał

```
JESTEM NAJMADRZEJSZY
```

Ale zrobiłeś błąd i napisałeś instrukcję

```
PRINT [JESTEM NAJGLUPSZY]
```

W takim razie nie naciskaj klawisza RETURN, ale DELETE BACK S tak długo, aż na ekranie zostanie tylko

```
PRINT [JESTEM NAJ■
```

Teraz dopisz linię do końca, ale już poprawnie.

```
PRINT [JESTEM NAJMADRZEJSZY]
```

Naciśnij RETURN i Twoja instrukcja zostanie wykonana.

```
JESTEM NAJMADRZEJSZY
```



DELETE BACK S jest jednym z klawiszy umożliwiających zmianę już napisanego tekstu programu, bez przepisywania go od początku. Opis działania innych klawiszy użytecznych przy redagowaniu tekstu znajduje się w następnych częściach tego podręcznika.

Posługując się LOGO można pisać na ekranie dowolne inne teksty poprzez danie instrukcji PRINT i odpowiedniego napisu ujętego w nawiasy kwadratowe "[", "]" tak jak to zilustrowano w poprzednim przykładzie.

CZYSZCZENIE EKRANU

Użyteczną komendą jest CT (od ang. Clear Text), czyści ona ekran i prznosi kursor na początek pierwszej linii.

CT Naciśnij RETURN

Stanie się dokładnie to, co opisaliśmy wyżej.

TWORZENIE PROCEDUR

Poprzez definiowanie nowych procedur możesz w LOGO tworzyć nowe komendy. W celu zasygnalizowania początku pisania procedury użyj komendy TO. W tej samej linii podaj nazwę procedury (nazwą może być dowolny ciąg znaków graficznych nie zawierających spacji).

Naciśnij RETURN. Następnie określ działanie procedury.

Dla przykładu zdefiniuj procedurę WITAJ w ten sposób, żeby komputer na każdorazowe napisanie witaj odpowiadał DZIEN DOBRY. JAK SIĘ MASZ?

Napisz:

TO WITAJ (słowa należy oddzielić spacją) Naciśnij RETURN

Znaczek zachęty "?" zmieni się na ">". Przypomina Ci on, że definiujesz teraz procedurę i wprowadzane komendy nie są realizowane na bieżąco lecz dołączone do definicji

```
> PRINT [DZIEN DOBRY]
> PRINT [JAK SIE MASZ]
> END
```

Naciśnij RETURN po każdej linii



END sygnalizuje koniec definiowania procedury.
LOGO odpowie:

```
WITAJ DEFINED
? █
```

Jako znak zachęty znowu pojawi się "?".
Wywołaj teraz procedurę poprzez podanie jej nazwy

```
WITAJ                Naciśnij RETURN
```

LOGO odpowie:

```
DZIEN DOBRY
JAK SIE MASZ?
? █
```

Jeżeli LOGO nie chce wykonać procedury prawdopodobnie coś źle wykonałeś. Później nauczysz się jak wywołać definicję procedury i jak ją poprawić. Tymczasem spróbuj napisać ją jeszcze raz, ale pod inną nazwą, na przykład WITAJ 1:

```
TO WITAJ 1
> PR [DZIEN DOBRY]
> PR [JAK SIE MASZ]
> END                Naciśnij RETURN
```

Wywołaj procedurę pisząc

```
WITAJ 1
```

LOGO odpowie:

```
DZIEN DOBRY
JAK SIE MASZ?
? █
```

Komendą REPEAT możesz kilkakrotnie wywołać tę samą procedurę. Niżej podajemy przykład użycia REPEAT w celu pięciokrotnego powtórzenia procedury WITAJ



REPEAT 5 [WITAJ]

(Nie zapomnij o spacji pomiędzy REPEAT i 5. Naciśnij RETURN).
LOGO napisze w odpowiedzi:

DZIEŃ DOBRY
JAK SIE MASZ?
DZIEŃ DOBRY
JAK SIE MASZ?
DZIEŃ DOBRY
JAK SIE MASZ?
DZIEŃ DOBRY
JAK SIE MASZ?
DZIEŃ DOBRY
JAK SIE MASZ?
?
■

Jeśli zechcesz komputer może Cię witać tysiąc razy:

REPEAT 1000 [WITAJ] Naciśnij RETURN

Pracę komputera możesz przerwać w dowolnym momencie naciskając BREAK. LOGO wtedy odpowie:

STOPPED! IN WITAJ
?
■

UWAGA

W następnych rozdziałach nie będziemy Ci już przypominać o naciśnięciu RETURN oraz oddzielaniu słów spacjami.

SŁOWNICZEK

W tym rozdziale wprowadziliśmy następujące słowa kluczowe (w nawiasach podane są skróty, których możesz używać zamiast pełnych form):

PRINT (PR)

CT

TO

END

REPEAT

Używaliśmy też specjalnych klawiszy:

[

]

DELETE BACK S

BREAK

RETURN



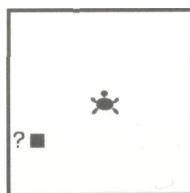
ROZDZIAŁ III

"SPOTKANIE Z ŻÓŁWIEM"

W tej części nauczysz się programować rysunki używając komputerowego stworka nazywanego żółwiem.

Pierwszy tego rodzaju żółw był robotem, wyglądał jak duża puszka i poruszał się na kółkach. Był sterowany przez komputer przy pomocy długiego kabla. Mógł rysować linię na podłodze, która była w tym celu przykrywana arkuszem papieru.

Nasz żółw żyje na ekranie telewizora. On również ma pisak, którym może kreślić linie. Bardzo dużo komend w LOGO służy właśnie do kierowania żółwiem. W tym rozdziale poznasz kilka najważniejszych.



Najpierw podaj komendę ST. Na środku ekranu pojawi się mały żółw. Ma on taki kształt, że widać dokładnie gdzie się znajduje i w którą stronę jest skierowana jego głowa. Tę pozycję i zwrot będziemy nazywać stanem żółwia. W każdym momencie jest on ściśle określony. Najważniejsze dla żółwia komendy dotyczą zmiany jego stanu.

W pierwszej chwili żółw znajduje się na środku ekranu i patrzy do góry. Zauważ, że teraz znak zachęty i kursor pojawiają się na dole. Twoje rozkazy będą wypisywane w pięciu ostatnich liniach ekranu.

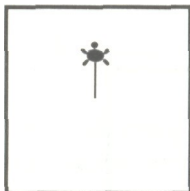


ZMIANA STANU ŻÓŁWIA FORWARD

Spróbujmy poruszyć żółwia używając komendy FORWARD. Jest to instrukcja wymagająca wprowadzenia pewnej liczby jako parametru. W tym przypadku parametr oznacza ilość kroków, które żółw ma wykonać.

Napisz następującą komendę:

```
FORWARD 50
```



Zauważ, że żółw zmienia pozycję zachowując poprzedni kierunek. Oczywiście jako parametr możesz w miejsce 50 podać inne liczby.

Odstęp pomiędzy komendą, a parametrem / w naszym przykładzie między FORWARD a 50 jest bardzo ważny. Pozwala komputerowi odróżnić słowo FORWARD od słowa FORWARD50. Większa ilość spacji oddzielających słowa, traktowana jest przez LOGO tak samo jak spacja pojedyncza.

Przy współpracy z LOGO przytrafiają się pewne błędy. Przeważają wśród nich prawdopodobnie pomyłki typu literowego.

Być może ich cechą wspólną jest brak odstępu pomiędzy komendami i ich parametrami?

Na przykład FORWARD jest w LOGO słowem kluczowym. Użyte jako komenda wymaga podania liczby parametru. FORWARD 50 jest pełnym rozkazem i powoduje przesunięcie żółwia naprzód. Natomiast FORWARD50 jest istotnie różne od FORWARD i nie jest zdefiniowane w LOGO (o ile nie zrobisz tego samodzielnie).

Dwie instrukcje różnią się tylko odstępem między słowami. Błędy wynikające z przestawienia lub pominięcia liter, choć trudne do zauważenia dla człowieka, stanowią istotną przeszkodę dla pracy komputera.

Jeśli napiszesz:

```
FRWARD
```



LOGO odpowie:

I DON'T KNOW HOW TO FRWARD

Zwracaj więc baczną uwagę na poprawność wprowadzanych napisów.

RIGHT

Chcąc zmienić kierunek żółwia podaj komendę **RIGHT** wraz z liczbą stopni o jaką chcesz go obrócić. W poniżej podanym przykładzie żółw przekreśli się w prawo o 90° .

RIGHT 90



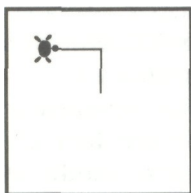
Żółw obraca się w prawo o kąt 90° w stosunku do poprzedniego kierunku. Zauważ, że zmianie ulega tylko kierunek, pozycja pozostaje ta sama.

BACK

Back posuwa żółwia do tyłu. Ta komenda, podobnie jak **FORWARD** zmienia tylko pozycję.

Na przykład:

BACK 50

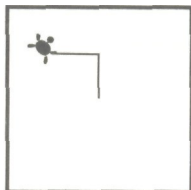


Także i tym razem 50 jest liczbą przykładową. Ty możesz podać inną.

LEFT

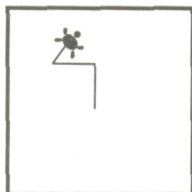
Ta komenda jest zupełnie podobna do RIGHT, ale obraca żółwia w przeciwnym kierunku.

LEFT 45



Żółw obraca się o 45° w swoją lewą stronę zmieniając kierunek i zachowując pozycję. Żeby to zobaczyć wyraźniej *rozkaz* jeszcze:

FORWARD 25



CS

Możesz eksperymentować komendami zmiany stanu zupełnie dowolnie. Możesz także użyć CS (od ang. Clear Screen) w celu wyczyszczenia ekranu i umieszczenia żółwia na środku. Teraz można zabawę rozpocząć od początku.



SŁOWNICZEK

W tym rozdziale zapoznałeś się z następującymi słowami kluczowymi: (w nawiasach podane są skróty)

BACK (BK)

CS

FORWARD (FD)

LEFT (LT)

RIGHT (RT)

ST



ROZDZIAŁ IV

"JAK NAUCZYĆ ŻÓŁWIA RYSOWAĆ KWADRAT"

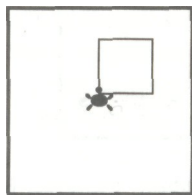
Używając komend z poprzedniego rozdziału możemy przy pomocy żółwia narysować kwadrat. Rozkazy zapisywać będziemy w wersji skróconej:

```
FD 30  
RT 90
```

```
FD 30  
RT 90
```

```
FD 30  
RT 90
```

```
FD 30  
RT 90
```



Zamieniając 30 na 50 dostaniesz większy kwadrat. (Możesz zresztą użyć dowolnej liczby).

Teraz zdefiniuj w LOGO nową komendę służącą do rysowania kwadratu. Będzie to procedura której wywołanie zastąpi żmudne wypisywanie poszczególnych instrukcji.

Aby zdefiniować procedurę, należy najpierw wybrać dla niej nazwę. Tę nazwij KWADRAT, gdyż wydaje się to naturalne (ale każde inne słowo będzie z punktu widzenia komputera równie dobre). Teraz użyj komendy TO i zdefiniuj KWADRAT podobnie jak poprzednio WITAJ.

To znaczy:

```
napisz TO KWADRAT, następnie odpowiednie instrukcje i zakończ słowem END.
```

Jest to dobra metoda, ale musisz pisać uważnie i bezbłędnie.



WPROWADZENIE DO ATARI LOGO EDITOR

Do redagowania procedury prowadzi także inna droga. Jest to ATARI LOGO EDITOR. Korzystając z niego można łatwo poprawiać popełnione błędy. Używanie Edytora ma także wady. Twoje rysunki wykonane z pomocą żółwia zostaną zastąpione ekranem Edytora i w efekcie niszczone. Jednak z drugiej strony używanie TO do definiowania procedur wymaga dużej koncentracji przy wprowadzaniu tekstu gdyż błędy można poprawić tylko w linii, w której znajduje się kursor.

EDIT (ED)

Sygnalizuje LOGO chęć wywołania Edytora. Komenda ta powinna zawierać znak " i nazwę procedury, którą chcesz zredagować. Nie wstawiać znaku spacji między " i nazwę procedury.

EDIT "KWADRAT



Po wprowadzeniu instrukcji klawiszem RETURN możesz używać ATARI LOGO EDITOR i tylko funkcję Edytora będą przez komputer wykonywane.

Jeśli zapomnisz poprzedzić nazwę procedury cudzysłowem (") napisz przykładowo:

EDIT KWADRAT

LOGO nie zrozumie Twoich intencji i odpowie:

I DON'T KNOW HOW TO KWADRAT

W momencie wywołania Edytora znikną z ekranu wszystkie dotychczasowe rysunki. Linia tytułowa z nazwą procedury pojawi się na górze ekranu.





Wyjaśnijmy znaczenie poszczególnych słów jeszcze raz na naszym przykładzie:

- TO KWADRAT - jest to linia tytułowa.
- TO - informuje, że następujący poniżej tekst będzie definicją procedury.
- KWADRAT - jest nazwą procedury zależną wyłącznie od inwencji piszącego program.

Kursor zaczyna linię tytułową. Zauważ, że LOGO nie wyświetla żadnego znaku zachęty podczas używania Edytora. Ponieważ nie chcesz zmieniać linii tytułowej wciśnij kombinację klawiszy CTRL+↓. Kursor przeskoczy do następnej linii. Teraz napisz instrukcję rysowania kwadratu. Są to komendy używane już poprzednio.



```
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
```

Wszelkie napisy w Edytorze traktuj jak ciągi znaczków na ekranie. Jeśli chcesz do któregoś z ciągów coś dodać przesuń w odpowiednie miejsce kursor. Używaj w tym celu kombinacji klawiszy: CTRL+← gdy chcesz go cofnąć, CTRL+→ gdy chcesz go przesunąć do przodu. Następnie napisz żądany znak, a zostanie on włączony do napisu.

Jeśli zrobiłeś błąd w poprzedniej linii, naciskaj CTRL+← tak długo, aż

kursor dojedzie do miejsca, w którym chcesz dokonać zmiany. Znaki, przez które przejeżdża kursor pozostają niezmienione. Chcąc skasować znak, nasuń na jego miejsce kursor i naciśnij CTRL DELETE BACK S. Znak pod kursorem zniknie.

Zauważ, że jeśli kursor jest na końcu napisu, naciskając CTRL DELETE BACK S można przesuwając następną linię tekstu tak długo aż zostanie ona dołączona do końca napisu, za którym stoi kursor.

Na przykład możesz z napisów:

```
FD 30 ■  
RT 90
```

uzyskać:

```
FD 30 ■ RT 90
```

W celu ponownego rozdzielenia linii naciśnij RETURN:

```
FD 30  
■ RT 90
```

Jeśli przypadkowo naciśniesz SPACE BAR zamiast RETURN ujrzysz na ekranie:

```
FD 30 ■ RT 90
```

Naciskając CTRL← odpowiednią ilość razy możesz przesunąć kursor, aż do litery T w słowie TO w linii tytułowej.

Gdy skończysz redagowanie, napisz END i wciśnij ESC. Jeżeli zapomnisz napisać słowo END zostanie ono wstawione przez LOGO automatycznie w momencie wciśnięcia ESC. Potem LOGO zamelduje:

```
KWADRAT DEFINED
```

UŻYCIE NOWEJ KOMENDY

Wypróbuj nową komendę pisząc KWADRAT. Żółw narysuje na ekranie kwadrat. Spróbuj jeszcze raz. Tym razem żółw obejdzie kwadrat, ale nie dorysuje żadnych nowych linii.

Obróć żółwia w lewo lub prawo i znowu użyj KWADRAT. Na ekranie otrzymasz nowy rysunek nałożony na poprzedni.

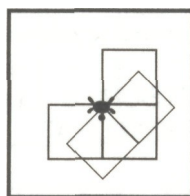
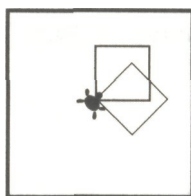
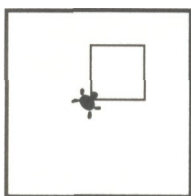
Na przykład powiedz żółwiowi:

```
RIGHT 45  
KWADRAT
```

Powtarzaj te dwie komendy. Możesz to zrobić używając pętli REPEAT. Wymaga ona podania dwóch parametrów. Na przykład:

```
REPEAT 3 [RT 45 KWADRAT]
```

Pierwszy parametr wskazuje ilość obiegów pętli, drugi określa powtarzaną instrukcję. Musi ona być ujęta w nawiasy kwadratowe. Wypróbuj podany przykład.



Stwórz teraz nową procedurę, która otrzyma nazwę GWIAZDA. Użyj do tego Edytora:

```
EDIT " GWIAZDA
```

Na ekranie pojawi się napis:

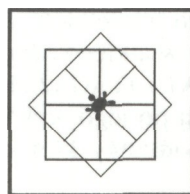
```
TO GWIAZDA
```

Kursor znajduje się na literze T w linii tytułowej. Ponieważ nie chcesz zmieniać, więc przejdź do następnej używając CTRL* i napisz w niej:

```
REPEAT 8 [KWADRAT RT 45]
```

i w nowej linii

```
END
```



Nie zapomnij o naciśnięciu ESC.

Zechcesz teraz zapewne wypróbować nową komendę. Umieść więc żółwia w stanie początkowym, jak pewnie pamiętasz służy do tego CS:

```
CS      i zaraz potem
GWIAZDA
```

Jeżeli przeszkadza Ci żółw w środku ekranu, możesz go usunąć wydając polecenie:

```
HT
```

Pamiętaj, że zawsze możesz go ponownie wywołać poleceniem ST.

Jeżeli Twoje kwadraty nie mają całkiem równych boków, to winę za to ponosi telewizor, a nie komputer ani LOGO. Możesz spróbować poprawić sytuację używając komendy SETSCR:

```
.SETSCR .8      i potem
CS KWADRAT
```

Jeżeli jest jeszcze gorzej użyj:

```
.SETSCR 1
```

Wypróbuj także inne parametry (również niecałkowite) i wybierz ten, który Cię zadowoli.

INNE SPOSOBY UŻYCIA PROCEDURY KWADRAT

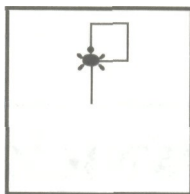
Teraz, kiedy już zdefiniowałeś znaczenie słowa KWADRAT możesz go używać tak, jak każdego słowa kluczowego w LOGO (czyli np. FD, BK, LT, RT, itd.). Co za tym idzie, nowe słowo może być przez Ciebie używane w definicjach nowych procedur. Fakt ten jest jedną z przesłanek stanowiących o wielkich możliwościach języka LOGO. Można podać wiele przykładów takiego użycia słowa KWADRAT.

FLAGA1 i FLAGA2 służą do tworzenia tego samego rysunku, ale pozostawiają po jego wykonaniu żółwia w różnych stanach. W obu procedurach końcowy kierunek żółwia jest ten sam, co na początku. Po

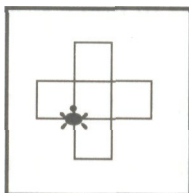


wykonaniu FLAGA1 żółt kończy pracę gdzie indziej niż ją zaczął. W procedurze FLAGA2 wraca on na swoje miejsce. Konsekwencje tej różnicy możemy obejrzeć w procedurach KRZYŻ i FLAGI. KRZYŻ polega na czterokrotnym wykonaniu FLAGA1, zaś flagi - na tyłokrotnym wykonaniu FLAGA2.

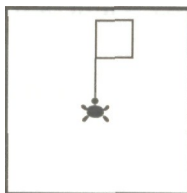
TO FLAGA
FD 30
KWADRAT
END



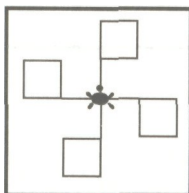
TO KRZYZ
REPEAT 4 [FLAGA RT 90]
END



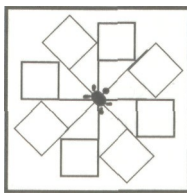
TO FLAGA2
FLAGA
BK 30
END
BK 30
END



TO FLAGI
REPEAT 4 [FLAGA2 RT 90]
END



TO DUZO FLAG
FLAGI
RT 90
FLAGI
END



Jeżeli teraz wyłączysz swój komputer, a potem włączysz go ponownie - po Twoich procedurach nie będzie śladu. Wszystkie zostaną zniszczone.

W następnym rozdziale dowiesz się, jak możesz przechowywać swoje procedury i stosować je ponownie, bez powtórnego wpisywania z klawiatury.

SŁOWNICZEK

Znajdziesz tu nowe słowa kluczowe LOGO, które pojawiły się w tym rozdziale:

EDIT (ED)
HT
.SETSCR

Używaliśmy też w nim pewnych specjalnych klawiszy lub ich kombinacji:

BREAK
CTRL←
CTRL→
CTRL↑
CTRL↓
CTRL DELETE BACK S
ESC

ROZDZIAŁ V

"ZAPISYWANIE I ODTWARZANIE PROGRAMÓW"

Nowo zdefiniowana procedura LOGO umieszczana jest w pamięci operacyjnej. Jest to część pamięci komputera zachowywana tylko wtedy, gdy jest on podłączony do prądu. Swoją procedurę możesz jednak utrwalić na kasecie magnetofonowej. Umożliwia to komenda SAVE. Przeciwnie działająca komenda LOAD odtwarza z kasyty uprzednio tam nagraną procedurę.

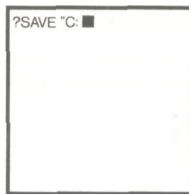
ZAPISYWANIE I ODTWARZANIE PROGRAMÓW

Podczas zapisywania programu cała zawartość pamięci operacyjnej komputera zostaje utrwalona na kasecie.

Przystępując do zapisywania powinieneś włożyć czystą kasetę do magnetofonu i ustawić taśmę na sam początek części magnetycznej (a nie rozbiegówki) i ustawić licznik magnetofonu na zero. Następnie należy napisać:

SAVE "C:

Nacisnij RETURN



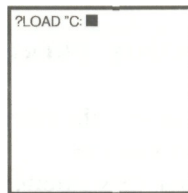
C: jest dla komputera sygnałem, że chcesz nagrywać na kasecie. Po wykonaniu tej czynności usłyszysz podwójny beep. W ten sposób komputer przypomina Ci, że masz na magnetofonie wcisnąć klawisze: PLAY i RECORD. Potem naciśnij jeszcze raz RETURN. Magnetofon sam ruszy. Sygnałem zakończenia nagrywania jest ponowne pojawienie się znaku zachęty i kursora oraz zatrzymanie się magnetofonu. Możesz go już teraz wyłączyć. To samo możesz zrobić z komputerem bez obawy zniszczenia swojej pracy. Na jednej stronie kasyty możesz przechowywać kilka

zbiorów (tak się nazywa to, co powstaje przy jednorazowym nagraniu. Dobrze jest więc wobec tego zapisywać stany licznika na początku i na końcu każdego zbioru. Rozsądnie jest także robić pomiędzy poszczególnymi zbiorami odstępy ok. 10 jednostek. W celu odtworzenia zbioru (czyli ponownego wpisania procedur z kasety do pamięci komputera musisz umieścić kasetę w magnetofonie, przewijając ją tak, by była ustawiona dokładnie w miejscu, w którym się zaczyna zbiór (ewentualnie kilka jednostek wcześniej).

Teraz napisz:

LOAD "C:

Naciśnij RETURN



Po usłyszeniu beepa wciśnij klawisz PLAY na magnetofonie i RETURN na klawiaturze komputera. Magnetofon sam ruszy, a pojawiające się kolejno na ekranie nazwy procedur z adnotacją DEFINED będą dowodem, że komputer umieścił je w swojej pamięci. Gdy na ekranie pojawi się znowu znak zachęty i kursor, a magnetofon się zatrzyma będzie to oznaczać, że nagrywanie się zakończyło.

SŁOWNICZEK

W tym rozdziale poznałeś tylko dwa nowe słowa kluczowe:

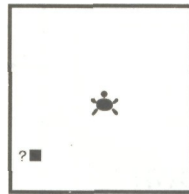
LOAD
SAVE

ROZDZIAŁ VI

"RYSUNKI I TEKST NA EKRANIE"

Przed podaniem LOGO pierwszej komendy dotyczącej żółwia, cały ekran przeznaczony jest na tekst. Od chwili, gdy podasz taką komendę, ekran będzie podzielony na dwa obszary - duży, przeznaczony dla żółwia, mały, na którym pojawi się tekst - ma on tylko 5 linii. Wydanie rozkazu TS ponownie spowoduje przeznaczenie całego ekranu na tekst. SS jest komendą, która dzieli ekran na obszary: przeznaczony dla żółwia i na tekst. Żadna z nich nie wymazuje tego, co już napisałeś lub narysowałeś - tekst i rysunki istnieją na ekranie jednocześnie, a Ty tylko wybierasz to, co chcesz oglądać. Naciśnięcie pewnych kombinacji klawiszy wywołuje ten sam efekt. CTRL T zastępuje TS, CTRL S zastępuje SS. Komenda FS (lub kombinacja CTRLF) powoduje, że cały ekran jest przeznaczony na wykonanie rysunków. Odpowiednie kombinacje CTRL T, CTRL S i CTRL F wywołują efekt także podczas wykonywania procedur, czyli wtedy gdy użycie pełnej wersji jest niemożliwe.

Uwaga: W przypadku użycia Edytora wszystkie rysunki wykonane przez żółwia zostaną zamazane. Bezpośrednio po jego opuszczeniu jak też zaraz po uruchomieniu ATARI LOGO opisane wyżej kombinacje klawiszy nie będą działać zanim nie wydasz jakiegokolwiek komendy żółwiowi pełnym tekstem.



SŁOWNICZEK

Oto nowe słowa kluczowe i odpowiadające im kombinacje klawiszy:

FS CTRL F
SS CTRL S
TS CTRL T

ROZDZIAŁ VII

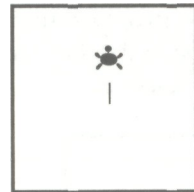
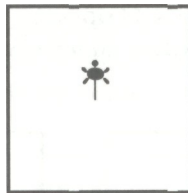
"PIÓRO ŻÓŁWIA, KOLOR RYSUNKÓW"

Żółw, gdziekolwiek się ruszy, pozostawia za sobą ślad. Dzieje się tak dlatego, że żółw posiada pióro. Jeżeli chcesz przemieścić żółwia tak, by nie pozostawiał za sobą śladu, musisz kazać mu podnieść pióro do góry. Możesz również kazać mu zmienić kolor atramentu, którym rysuje. Ten rozdział wyjaśni Ci, jak korzystać z tych właśnie możliwości Grafiki Żółwia ATARI LOGO.

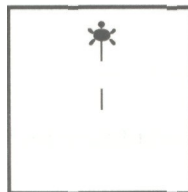
KOMENDY DOTYCZĄCE PIÓRA

Chcąc podnieść pióro, używaj komendy PENUP (PU), a opuścić - komendy PENDOWN (PD). Obejrzyj działanie tych komend na następującym przykładzie:

```
FD 15  
PENUP  
FD 15
```



```
PENDOWN  
FD 15
```



Podnoszenie i opuszczanie pióra będziemy nazywać zmianą stanu pióra. Są jeszcze dwie inne komendy zmieniające stan pióra. PE i PX. Po rozkazie PE pióro żółwia staje się gumką i wyciera wszystkie rysunki, na które natrafi. Znaczy to, że gdy przejdziemy żółwem po jego poprzednim śladzie, to zostanie on zmazany. Wypróbuj tę komendę na poniższym przykładzie:

```
CS
PENDOWN
KWADRAT
PE
KWADRAT
```

Żółw będzie wycierał wszystko, co napotka na swej drodze dopóki nie przerwiemy tego komendą PU lub PD. Zauważ również, że w tej sytuacji żółw niczego nie rysuje.

Stan wytwarzający się po komendzie PX jest połączeniem stanów PE i PD. Żółw wówczas rysuje linie wtedy, gdy porusza się po czystym podłożu, gdy natomiast trafia na już istniejące ślady - wyciera je. Dyrektywa ta może być użyta dla tworzenia szczególnie wyrafinowanych efektów. Oto (bardzo nieskomplikowany) przykład:

```
PX
KWADRAT
KWADRAT
KWADRAT
```



Jak zwykle PD powoduje powrót do stanu podstawowego.

UŻYCIE KOLORÓW W GRAFICE ATARI

W tym fragmencie opisane są efekty, które można uzyskać stosując kolory ATARI.

Komputer domowy ATARI pozwala Ci użyć 128 różnych kolorów.

Składa się na nie 16 kolorów zasadniczych, każdy w 8 odcieniach. Są one kodowane za pomocą liczb.

0 - 7	szary
8 - 15	jasnopomarańczowy (żółty)
16 - 23	pomarańczowy
24 - 31	pomarańczowo-czerwony
32 - 39	różowy
40 - 47	purpurowy
48 - 55	purpurowo-niebieski
64 - 71	granatowy
72 - 79	błękitny
80 - 87	turkusowy
88 - 95	niebiesko-zielony
96 - 103	zielony
104 - 111	żółto-zielony
112 - 119	pomarańczowo-zielony
120 - 127	jasno-pomarańczowy.

Kolory można uzyskać tylko na telewizorach systemu PAL. Zależą one ponadto od rodzaju użytego telewizora.

Dla każdego z kolorów zasadniczych najniższy numer oznacza odcień najciemniejszy, zaś numer najwyższy - najjaśniejszy. Np. 0 oznacza kolor czarny, a 7 - biały. Możesz dokonywać trzech rodzajów zmian koloru. Są to: zmiana koloru tła rysunku, koloru pióra, którym rysuje żółw. Możliwa jest też zmiana koloru samego żółwia (patrz Rozdział 16).

ZMIANA KOLORU TŁA

Kiedy ATARI LOGO rozpoczyna pracę, tło jest w kolorze nr 74 (niebieski). Do zmiany tego koloru służy komenda SETBG, a jej parametrem jest numer koloru, który pragniesz uzyskać.

```
SETBG 1
SETBG 40
SETBG 120
```

Możesz zdefiniować procedurę, która będzie cyklicznie zmieniała kolory tła. Abyś mógł je dokładniej obejrzeć wykorzystaj nową komendę - WAIT.

WAIT 50 każe LOGO odczekać 1 sekundę przed wykonaniem



następnego polecenia (czas oczekiwania jest proporcjonalny dla parametru - 1 i odpowiada 0.02 sekundy). Jako przykład jej użycia może służyć właśnie poniższa procedura:

```
TO KOLORY
SETBG 0 WAIT 20
SETBG 35 WAIT 20
SETBG 48 WAIT 20
SETBG 60 WAIT 20
SETBG 98 WAIT 20
SETBG 126 WAIT 20
END
```

Powtórz kolory kilkakrotnie. Aby to zrobić zapisz:

```
REPEAT 3 [KOLORY]
```

W każdym momencie możesz uzyskać informację, jaki kolor tła właśnie oglądasz. Napisz po prostu:

```
PRINT BG
```

Odpowiedzią LOGO będzie wypisanie liczby 126. Zmień teraz kolor na czarny:

```
SETBG 0
PRINT BG
```

ZMIANA KOLORU PIÓRA

Żółtów może rysować trzema różnymi piórami ponumerowanymi liczbami 0, 1 i 2. W momencie startu żółtów używa pióra o numerze 0 (odpowiada mu kolor złoty - nr 15 w spisie kolorów ATARI). Jeśli nie zmienisz pióra, to kolejne instrukcje spowodują narysowanie kwadratu w kolorze złotym:

```
CS
REPEAT 4 [FD 30 RT 90]
```

Możesz dowiedzieć się o numer pióra, którego używa żółtów rozkazem PN.



PRINT PN

Odpowiedź LOGO wygląda tak:

0

Zmiany pióra możesz dokonać komendą SETPN. Dzięki temu możesz teraz rysować w trzech różnych kolorach. Pióro nr 1 rysuje na purpurowo, zaś nr 2 na czerwono. Następującymi komendami możesz narysować kwadraty czerwony i purpurowy używając piór 1 i 2:

```
SETPN 1
REPEAT 4 [FD 30 RT 90]
LT 180
SETPN 2
REPEAT 4 [FD 30 RT 90]
```

Pamiętaj, że gdy zmienisz kolor tła, to kolory piór mogą przestać dokładnie odpowiadać numerom podanym w tabeli. Teraz podamy Ci sposób na to, by dowiedzieć się o numery kolorów, którymi rysują poszczególne pióra. Napisz:

```
PRINT PC 0
15
PRINT PC 1
47
PRINT PC 2
121
```

A teraz dowiesz się w jaki sposób zmieniać kolory piór. Robi się to używając komendy SETPC (od SET Pen Color). Każde ona zmienia kolor pióra oznaczonego numerem, który wprowadzasz jako pierwszy parametr na kolor, który jest zakodowany (wg. tabeli) drugim parametrem. Napisz:

```
SETPC 0 40
```

Już pewnie zauważyłeś, że zmiana koloru pióra pociąga za sobą zmianę koloru wszystkich narysowanych nim linii. Kwadrat, który był złoty, stał się karmazynowy. Natomiast inne kwadraty nie zmieniły swoich kolorów.

SETPN 0
FD 50

Jak widzisz żółw rysuje piórem "0" także na karmazynowo.

SŁOWNICZEK

Nowe słowa kluczowe to:

BG
PC
PE
PENDOWN (PD)
PENUP (PU)
PN
PX
SETBG
SETPC
SETPN
WAIT

ROZDZIAŁ VII

"NOWE INFORMACJE O UŻYCIU EDYTORA"

Edytor ATARI LOGO pozwala Ci zmieniać gotowe, już zdefiniowane procedury, równie dobrze jak tworzyć nowe. Możesz dokonywać niewielkich zmian w procedurach lub zmieniać im nazwy. Zacznij od zdefiniowania procedury rysującej diament, ale z błędem w tekście, aby potem przekonać się, jak można go poprawić. Na przykład zrób to tak:

```
TO DIAMENT  
KWADRAT  
RT 45  
END
```

Po uruchomieniu stwierdzisz, że ta procedura rysuje zwyczajny kwadrat, a nie diament. Powód jest oczywisty - rozkaz RT 45 powinien być użyty przed narysowaniem KWADRATU, a nie po.

Aby ten błąd poprawić wyświetl tekst procedury na ekranie pisząc:

```
EDIT " DIAMENT
```

LOGO spełni Twoje żądanie pisząc:

```
TO DIAMENT  
KWADRAT  
RT 45  
END
```

Kursor jest umieszczony w lewym górnym rogu ekranu na literze T w słowie TO.

Jeśli chcesz cokolwiek dopisać lub usunąć musisz przesunąć go w odpowiednie miejsce. Dokonasz tego za pomocą kombinacji klawiatury CTRL→ i CTRL←

W tym przypadku przesuń kursor do końca linii pierwszej (nic tu nie ma do zmiany) i potem naciśnij RETURN. W nowo utworzonej linii napisz RT 45 i potem dwukrotnym naciśnięciem CTRL→ przejdź do przedostatniej linii. Sytuacja będzie taka:



Teraz przez kilkukrotne naciśnięcie DELETE BACK S wymaż zbędną linię.

WYJŚCIE Z EDYTORA

Naciśnij klawisz ESC. Jest to dla komputera sygnał, że zakończyłeś redakcję tekstu. LOGO powiadomi Cię o tym, że procedura DIAMENT została zdefiniowana komunikatem DIAMENT DEFINED.

Jeśli pracując nad tekstem rozmyśliłeś się i chcesz cofnąć poprawki, których dokonałeś, przerwij pracę Edytora klawiszem BREAK. LOGO wychodząc z Edytora nie uwzględni dokonanych poprawek i tekst procedury nie zostanie zmieniony.

Niżej znajdziesz krótki opis funkcji użytecznych przy redagowaniu tekstu:

CTRL↑	- przesuwa kursor do następnej linii
CTRL↓	- przesuwa kursor do poprzedniej linii
CTRL←	- przesuwa kursor o jedną pozycję w prawo
CTRL→	- przesuwa kursor o jedną pozycję w lewo
RETURN	- Przesuwa kursor i znajdujący się za nim tekst do następnej, nowo utworzonej linii
* CTRL A	- przesuwa kursor na początek bieżącej linii
* CTRL E	- przesuwa kursor na koniec bieżącej linii
DELETE BACK S	- wymazuje znak na lewo od kursora
CTRL DELETE BACK S	- wymazuje znak pod kursorem
* SHIFT DELETE BACK S	- wymazuje znaki (wszystkie) znajdujące się za kursorem aż do końca linii.

Funkcje oznaczone * nie były omawiane, ale są wygodne i doradzamy Ci ich stosowanie.

Kursor może się poruszać po ekranie tylko tam, gdzie się znajduje tekst. Jeśli pomimo tego spróbujesz go przesunąć, ATARI LOGO wyrazi swoje niezadowolenie sygnałem dźwiękowym.

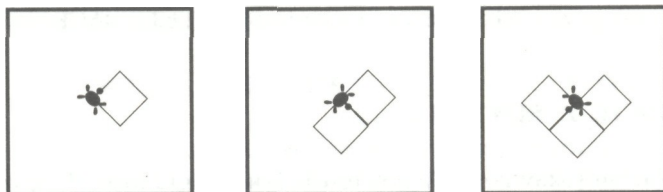
REDAGOWANIE POZA EDYTOREM

Większość z funkcji Edytora umiesz także wykorzystać do redagowania programów poza Edytorem.

Napisz np:



DIAMENT



Naciśnij teraz kombinację CTRL Y. Da ona kopię ostatnio przez Ciebie napisanej linii. Jak widzisz, każda linia na ekranie jest jakby mini-Edytorem, składającym się tylko z niej samej.

LOGO więc odpowie:

DIAMENT

Kursor znajduje się na końcu linii. Naciskając CTRL A przesunij go na początek i dopisz RT 45 oraz spację. Naciśnij RETURN. Spróbuj użyć innych funkcji edycyjnych, które działają poza Edytorem (to te oznaczone w spisie).

"PRZESTRZEŃ ROBOCZA"

Współpracując z komputerem nadajesz często słowom nowe znaczenie (np. definiując procedurę kwadrat). LOGO wkłada te nowe słowa (wraz z ich znaczeniem) do pewnej części pamięci komputera, którą dalej będziemy nazywać przestrzenią roboczą.

Jest kilka sposobów obejrzenia aktualnej zawartości przestrzeni roboczej. Możesz na przykład odczytać z pamięci komputera linie tytułowe wszystkich dotychczas zdefiniowanych procedur, jak również obejrzeć teksty dowolnie wybranych spośród nich albo wszystkich jednocześnie.

ODCZYTYWANIE PROCEDUR

Komenda POTS (Print Out TitleS) powoduje wypisanie na ekranie linii tytułowych wszystkich istniejących w przestrzeni roboczej procedur.

Napisz:

```
POTS
```

LOGO odpowiada:

```
TO DIAMENT  
TO KOLORY
```

itd

POPS (Print Out ProcedureS) drukuje całe definicje procedur:

```
POPS
```

LOGO odpowiada:

```
TO DIAMENT  
RT 45  
KWADRAT  
END  
TO KOLORY
```



```
SETBG 0 WAIT 20
```

```
.
```

```
.
```

```
itd
```

Możesz także wywołać z pamięci tekst jednej dowolnie wybranej procedury. Na przykład:

```
PO "GWIAZDA
```

LOGO odpowie:

```
TO GWIAZDA  
REPEAT 8 [KWADRAT RT 45]  
END
```

Instrukcją PO można wywołać tekst więcej niż jednej procedury podając jako parametr listę nazw, np. komenda:

```
PO [KWADRAT GWIAZDA DIAMENT]
```

spowoduje wypisanie definicji trzech procedur. Pamiętaj o możliwości wykorzystania TS i CTRL T - wtedy napisany tekst będziesz mógł oglądać na całym ekranie.

PRZESUWANIE PROCEDUR Z PRZESTRZENI ROBOCZEJ

Przestrzeń roboczą można opróżnić poprzez likwidowanie poszczególnych procedur (robi się to, gdy zaczyna brakować miejsca na nowe procedury lub w celu usunięcia procedur nieudanych). Wymaga to zawsze zastanowienia, gdyż jest to czynność nieodwracalna i o ile nie zapiszesz usuwanych procedur na taśmie, przy ponownym użyciu będziesz je musiał definiować na nowo. Niżej podaliśmy kilka przykładów komend wymazujących procedury z przestrzeni roboczej. Parametrem pierwszej z nich ERASE (forma skrócona ER), jest nazwa lub lista nazw procedur, które chcesz zniszczyć.

```
ERASE "DIAMENT
```

usuwa z pamięci procedurę DIAMENT, zaś



ERASE [KWADRAT, GWIAZDA]

procedury KWADRAT i GWIAZDA.

Komenda ERPS wymazuje od razu całą zawartość przestrzeni roboczej.

SŁOWNICZEK

W tym rozdziale poznałeś następujące słowa kluczowe:

POTS

POPS

PO

ERASE (ER)

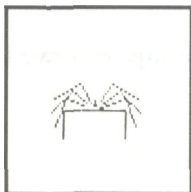
ERPS



ROZDZIAŁ X

"PROJEKT PROCEDURY RYSUJĄCEJ PAJĄKA"

Proponujemy Ci stworzenie rysunku pająka mającego z każdej strony 4 nogi.

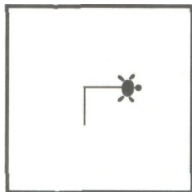


Najpierw przyjrzyj się dokładnie lewej i prawej nodze. Na każdą kończynę składają się dwa odcinki schodzące się pod kątem prostym.

W pierwszym kroku zdefiniuj prawą nogę:

```
TO PRAWANOGA  
FD 30  
RT 90  
FD 30  
END
```

i wypróbuj ją.

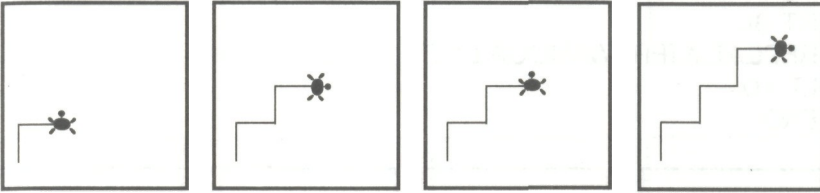


Procedura rzeczywiście rysuje prawą nogę, ale żółw zatrzymuje się w miejscu bardzo niedogodnym do rysowania następnej kończyny. Ta procedura świetnie nadaje się do rysowania schodów.

```
LT 90
```



PRAWA NOGA
LT 90
PRAWANOGA



Tobie jednak chodziło o pająka. Wniosek jest taki, że żółw po wykonaniu procedury powinien znaleźć się w punkcie, z którego wyszedł. Skorzystaj z Edytora:

EDIT "PRAWANOGA

LOGO udostępni Ci procedurę z kursorem umieszczonym na T:

```
TO PRAWANOGA
FD 30
RT 90
FD 30
END
```

Umieść odpowiednio kursor i dopisz nowe komendy:

```
BK 30
LT 90
BK 30
END
```

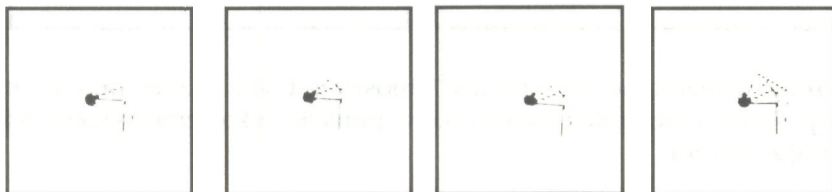
Wypróbuj teraz nową procedurę.

Teraz zaprojektuj procedurę, która będzie rysowała wszystkie prawe nogi pająka:

```
CS
RT 90
PRAWANOGA
LT 20
PRAWANOGA
```

Teraz już wiesz jak to należy zrobić. Napisz więc gotowy tekst procedury:

```
TO PRAWASTRONA
RT 90
REPEAT 4 [PRAWANOGA LT 20]
LT 10
END
```



Zauważ, że ostatnia komenda w procedurze PRAWASTRONA, LT 10, przywraca żółwia do tej samej pozycji, w jakiej znajdował się on w momencie startu.

Pozostawianie żółwia w stanie w jakim go zastaliśmy w momencie startu jest dobrym zwyczajem programowania.

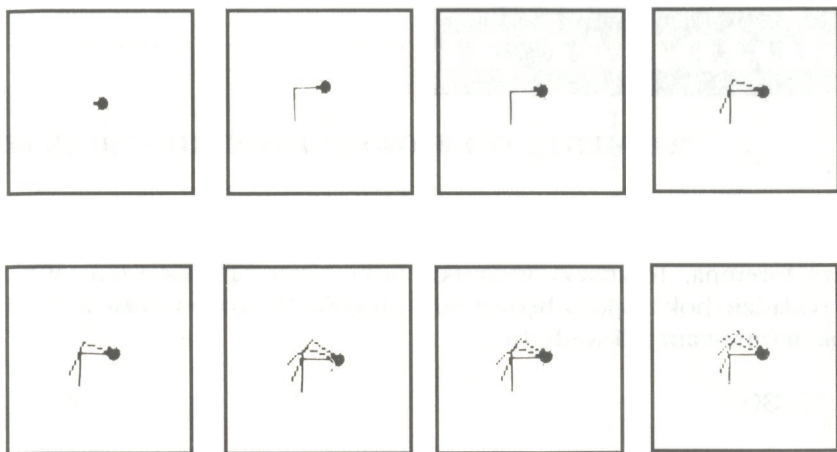
Teraz wykonaj lewą nogę. Procedura jest bardzo podobna do tej którą napisałeś definiując prawą nogę:

```
TO LEWANOGA
FD 30
LT 90
FD 30
BK 30
RT 90
BK 30
END
```

Wypróbuj to, następnie użyj do zdefiniowania procedury LEWASTRONA:

```
TO LEWASTRONA
LT 90
REPEAT 4 [LEWANOGA RT 20]
RT 10
END
```





I wreszcie możesz zdefiniować całego pająka:

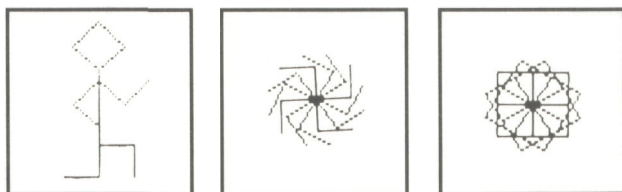
```

TO PAJĄK
LEWASTRONA
PRAWASTRONA
FD 10 BK 10
RT
END

```



Używając procedur PRAWANOGA i LEWANOGA możesz rysować także inne rysunki.



ROZDZIAŁ XI

"ELEMENTY ŻÓLWIOWEJ GEOMETRII - TRÓJKĄTY"

Żółw potrafi rysować różne trójkątne kształty. Trójkąt o którym będziemy mówić poniżej jest podobnie jak omawiany wcześniej kwadrat figurą foremną. To znaczy wszystkie boki i kąty są takie same. W tym przykładzie bok trójkąta będzie miał długość 30 kroków żółwia (tak jak bok definiowanego kwadratu).

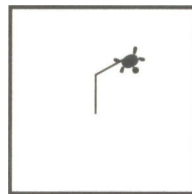
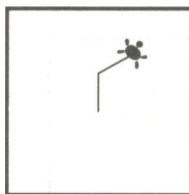
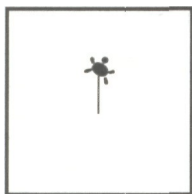
FD 30

Teraz chwila zastanowienia przed podjęciem decyzji. O ile stopni powinien obrócić się żółw, aby narysować trójkąt? Ze szkoły wiesz, że w trójkącie równobocznym każdy kąt ma 60° . Co się jednak stanie gdy obrócimy naszego żółwia o 60° :

RT 60

FD 30

RT 60



Interesujące, ale to nie jest trójkąt! Skoro jednak zacząłeś dokończ robotę:

FD 30

RT 60

FD 30

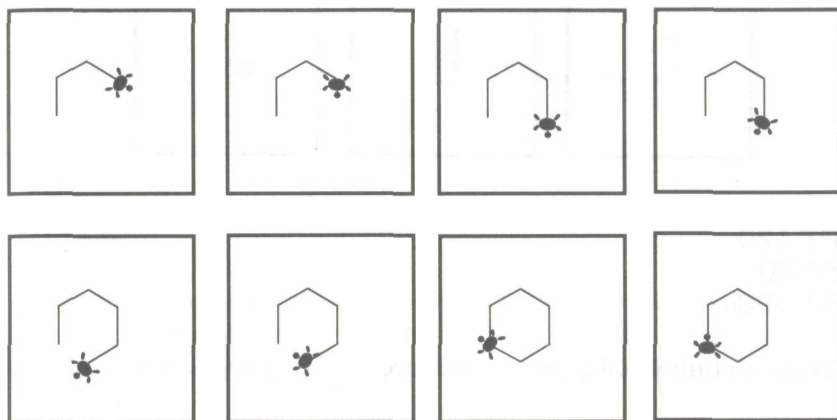
RT 60

FD 30

RT 60

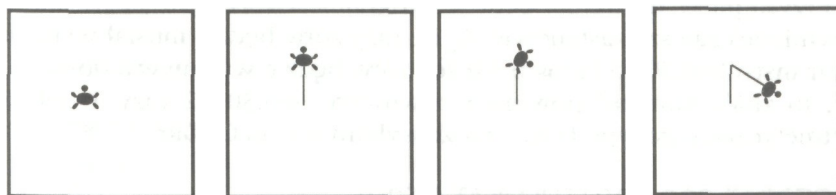
FD 30

RT 60



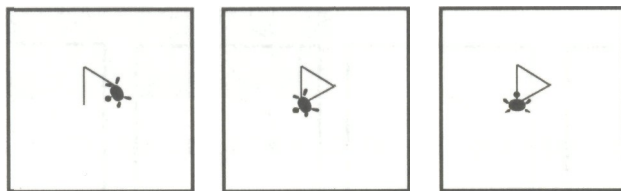
Żółw wykreślił sześciokąt - figurę o 6 bokach zamiast oczekiwanych 3. Żeby otrzymać trójkąt żółwia należy obracać o 120° , na każdym rogu. Pytasz dlaczego? Odpowiedź jest prosta. Spójrz na problem oczyma żółwia wykonującego rysunek. Od momentu startu do chwili zakończenia żółw musi się obrócić o pełny kąt czyli o 360° , aby wylądować w takiej samej pozycji. Idzie on prosto wzdłuż boków trójkąta i obraca się w wierzchołkach (niekoniecznie o kąt, który ma być kątem wewnętrznym trójkąta). Są trzy wierzchołki, a więc trzy obroty żółwia - w sumie mają dać pełny obrót o 360° , ($3 \cdot 120^\circ$) więc widać stąd, że za każdym razem żółw musi się obracać o 120° . Rysując sześciokąt obracał się $6 \cdot 60^\circ$, zaś kwadrat 4 razy po 90° ($4 \cdot 90 = 360$).

Teraz możemy dokończyć rysunek trójkąta:



CS
FD 30
RT 120
FD 30





```
RT 120
FD 30
RT 120
```

Teraz zdefiniuj trójkąt jako procedurę. Wykorzystaj Edytor:

```
EDIT TROJKAT
```

Wprowadź linie:

```
REPEAT 3 [FD 30 RT 120]
END
```

Pobaw się przez moment tą procedurą. Np:

```
REPEAT 3 [TROJKAT RT 120]
REPEAT 6 [TROJKAT RT 60]
REPEAT 100 [TROJKAT RT 30]
```

W ostatnim przykładzie żółw powtarza swoją drogę wiele razy. Jego działalność możesz zawsze przerwać naciskając **BREAK**.

Przystępując do użycia pętli w programie tworzącym jakiś rysunek powinieneś zawsze zastanowić się ile razy żółw będzie musiał wykonać zbiór instrukcji. Przykładowo - jeżeli żółw będzie wykonywał obroty po 30° , to zbiór komend powinien powtórzyć $360:30=12$ razy. Działania arytmetyczne tego typu LOGO może wykonywać za Ciebie:

```
REPEAT 360 / 30 [TROJKAT RT30]
```

Niżej przedstawiamy Ci kilka propozycji rysunków, które można wykonać przy użyciu procedury TROJKAT.

Proponujemy Ci najpierw narysować namiot. Samo wywołanie procedury TROJKAT tutaj nie wystarczy. Namiot będzie przewrócony.



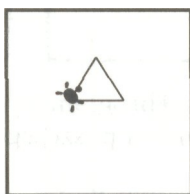
CS
TROJKAT

Obróć żółwia o 90° w prawo i teraz spróbuj wywołać TROJKAT. Tym razem też się nie udało. Namiot stoi do góry nogami.

CS
RT 90
TROJKAT

Przypomnij sobie jak rysowaliśmy TROJKAT. Kąt wewnętrzny trójkąta miał 60° . Jeśli teraz obrócimy żółwia o 90° w prawo a następnie w lewo o 60° (LT 60). W tej chwili jest on gotów narysować namiot:

CS
RT 30
TROJKAT



Cała procedura wygląda więc:

TO NAMIOT
RT 30
TROJKAT
END

OD NAMIOTU DO DRZEWA

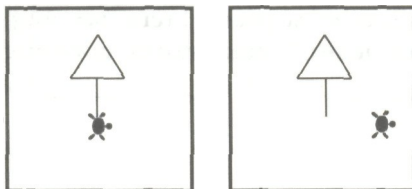
Zdefiniowany NAMIOT pomoże nam określić DRZEWO:

TO DRZEWO
NAMIOT
RT 60
FD 15
RT 90
FD 15
RT 180
END

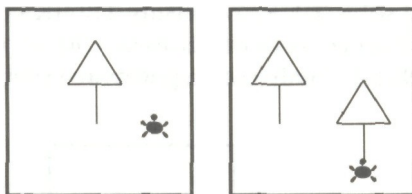


Teraz możesz nie tylko rysować jedno drzewo, ale cały las na ekranie.
Na przykład:

```
CS
DRZEWO
RT 90
PU
FD 30
```



```
LT 90
PD
DRZEWO
```



Ucz się zwyczajów dobrego programisty i umieszczaj instrukcje przygotowujące żółwia do wykonania poszczególnych fragmentów rysunku w oddzielnej procedurze.

Przykładowo zdefiniuj ODSZĘP. Będzie to procedura określająca działania żółwia pomiędzy rysowaniem kolejnych drzew.

```
TO ODSZĘP
RT 90
PU FD 30
LT 90 PD
END
```

Teraz możemy powtarzać na przemian: DRZEWO i ODSZĘP

```
REPEAT 3 [DRZEWO ODSZĘP]
```

Jeśli chcesz las zagęścić, zmniejszając przerwy między drzewami wywołaj Edytorem ODSZĘP i zmniejsz parametr instrukcji FORWARD przesuwając żółwia na przód.

ŻÓŁW BUDUJE DOM

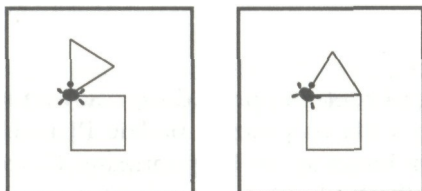
Nauczyłeś już żółwia rysować kwadrat i trójkąt. Złóż to teraz razem i zbuduj dom.



CS
KWADRAT
FD 30
TRÓJKAT

Złożyłeś kwadrat i trójkąt, ale żółw nie chce rysować domu!? Poppełni-
łeś błąd na szczęście łatwy do naprawienia. Zastąp trójkąt namiotem
i wszystko będzie w porządku:

CS
KWADRAT
FD 30
NAMIOT



ROZDZIAŁ XII

"ZMIENNE DUŻE I MAŁE KWADRATY"

Może się zdarzyć, że zechcesz rysować kwadraty o różnych długościach boków: 60, 50, 100, 10 itd. Zrobisz to z łatwością gromadząc wiele procedur KWADRAT 100, KWADRAT 50, KWADRAT 33 ...

Cel możesz osiągnąć także metodą o wiele krótszą uzależniając wielkość kwadratu od wprowadzonego parametru. W zależności od potrzeby będziesz teraz określać rozmiar figury pisząc:

KWADRAT 50

KWADRAT 33

KWADRAT 13

Zabierz się do pracy i zdefiniuj procedurę rysowania kwadratów różnej wielkości. Stosowną dla niej nazwą będzie PUDEŁKO, gdyż swoim kształtem kojarzy się z kwadratem. Proponujemy Ci jednak zmodyfikować nieco nazwę na PUDELKOP. Jest to skrót od nazwy - PUDEŁKO PRAWO, pozwalającej je odróżnić od pudełka lewego, które wkrótce określimy oczywiście pod nazwą PUDELKOL.

PUDELKOP możesz w szybki sposób zdefiniować przerabiając procedurę KWADRAT. Użyj w tym celu Edytora. Nie obawiaj się, że przeróbkami zniszczysz pierwotną procedurę KWADRAT. Jeżeli przed wyjściem z Edytora w pozmienianej definicji zmienisz także nazwę będziesz miał odtąd dwie procedury: starą - Kwadrat i nową - przerobiony KWADRAT (tak naprawdę w Edytorze pracujesz na kopii tekstu, którego pierwovzór znajduje się w pamięci komputera. Pierwovzór ten można oczywiście zniszczyć, ale tylko wtedy gdy wypiszesz nową procedurę pod starą nazwą).

Wywołaj więc KWADRAT Edytorem:

EDIT " KWADRAT

Komputer odpowie definicją procedury:

TO KWADRAT

FD 30

RT90



```
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
END
```

Zacznij od zmiany nazwy procedury. Cursor znajduje się na literze "T" w słowie TO, przesuń za słowo KWADRAT używając CTRL E. Teraz "wymaż" starą nazwę KWADRAT wciskając odpowiednią ilość razy DELETE BACKS i napisz nową nazwę PUDELKOP. Wciśnij ESC.

LOGO zamelduje:

```
PUDELKOP DEFINED
```

Pamiętaj - oryginalna definicja KWADRAT nadal istnieje. W tej chwili w pamięci komputera są dwie procedury złożone z tych samych instrukcji, ale mające inne nazwy.

Teraz możesz przerobić definicję PUDELKOP. Napisz:

```
EDIT " PUDELKOP
```

ATARI odpowie:

```
TO PUDELKOP
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
END
```

Zmień PUDELKOP tak, by móc tworzyć kwadraty różnej wielkości. Długość boku rysowanego kwadratu musisz uzależnić od parametru, który będziemy wprowadzać wywołując procedurę (zupełnie podobnie jak długość rysowanego przez żółwia odcinka zależy od liczby wprowadzonej jako parametr instrukcji FORWARD). Jak dojść do tego w LOGO?

Pierwszą instrukcją musi być FORWARD i liczba. Ale pisząc procedurę nie znasz wartości tej liczby, co więcej, chcesz aby "pasowała" do wszystkich wielkości kwadratów. Wybrniesz z tej opresji nadając owej nieznannej liczbie nazwę. Przykładowo niech będzie to BOK. Teraz gdy napiszesz FD :BOK będzie to dla żółwia oznaczać: idź na przód o tyle kroków, ile pokazuje liczba nazwana BOK). W chwili gdy żółw przystąpi do wykonywania tej instrukcji, BOK musi mieć oczywiście konkretną liczbową wartość, co nie zmienia faktu, że przy każdym wywołaniu procedury wartość ta może być inna).

Możesz więc definiować:

```
FD :BOK
RT 90
FD :BOK
RT 90
FD :BOK
RT 90
FD :BOK
Rt 90
```

Jeszcze jeden pomysł jest potrzebny, aby przemienić to w poprawną procedurę. Chcesz, żeby komenda PUDELKOP wymagała wprowadzenia liczby jako parametru, innymi słowy, żeby wywołanie jej wyglądało tak:

```
PUDELKOP 20
PUDELKOP 100
```

Nasz parametr nazywa się BOK.

Tłumacząc powyższe uwagi na język komputera w definicji piszemy:

```
TO PUDELKOP BOK
FD :BOK
RT 90
FD :BOK
RT 90
FD :BOK
RT 90
FD :BOK
RT 90
FD :BOK
RT 90
END
```

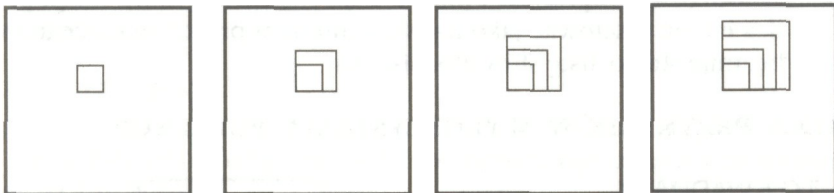


Naciśnij ESC. LOGO odpowie:

PUDELKOP DEFINED

Udało Ci się skonstruować procedurę sterującą żółwiem, tak żeby rysował kwadraty różnej wielkości, zależnej od wprowadzonego parametru:

PUDELKOP 10
PUDELKOP 20
PUDELKOP 30
PUDELKOP 40



Teraz potrafisz już używać do programowania mocnego aparatu matematycznego jakim są zmienne. Zupełnie tak samo jak w szkolnej algebrze, z tą tylko różnicą, że zamiast oznaczać zmienne przez X możesz nadawać im dowolne nazwy, najlepiej takie, które (jak nasz BOK) kojarzą się z tym co faktycznie niewiadoma przedstawia.

Przeanalizujmy nasze PUDELKOP jeszcze raz, by lepiej zrozumieć i utrwalić nowe wiadomości.

Przystępując do określenia procedury, chciałeś powiedzieć żółwiowi jak należy rysować kwadrat, nie wiedząc z góry jakiej wielkości figura będzie Ci potrzebna. Inaczej mówiąc, chciałeś nauczyć żółwia rysować dowolny kwadrat. Gdy przyszła kolej na napisanie instrukcji FORWARD wiedziałeś, że wymaga ona podania parametru. Nie mogłeś napisać samego FORWARD. Musiałeś napisać FORWARD "coś". Nadałeś temu "coś" nazwę BOK. Dla LOGO wyrażenie : BOK oznacza "staje się tym, co w kontenerze nazywa się BOK".) Kontenerem przyjęło się oznaczać ten fragment pamięci komputera, gdzie zapisywane są między innymi aktualne wartości zmiennych). Żeby LOGO wykonał komendę FORWARD: BOK kontener nie może być pusty. Kontener zapełnia się gdy wywołujesz komendę PUDELKOP podając jej nazwę wraz z parametrem:

PUDELKOP 10 lub PUDELKOP 15 itd.



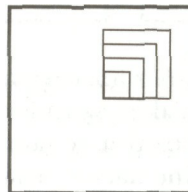
Gdy LOGO czyta tak napisaną komendę, liczba 10, 15 lub każda inna którą wprowadzisz na ich miejsce, jest zakładana do kontenera i nazywa się BOK. Komputer może wtedy "zaglądać" do kontenera w czasie wykonywania procedury.

Możliwe błędy:

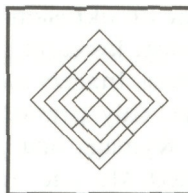
1. Błędy literowe w nazwie zmiennej (np. POK zamiast BOK), inna nazwa tej samej zmiennej w linii tytułowej, inna w instrukcjach.
2. Pominięcie: (dwukropek),
3. Umieściłeś spację między : (dwukropek) i BOK albo też : (dwukropek) przed liczbą,
4. Zapomniałeś o spacji, która powinna poprzedzać : (dwukropek).
Znak : (dwukropek) daje LOGO znać, że słowo, które on poprzedza ma on traktować jako nazwę kontenera przechowującego liczbę, inne słowo, listę słów albo listę list.

KILKA PRZYKŁADÓW WYKORZYSTANIA PUDELKOP

```
TO KWADRATY
PUDELKOP 10
PUDELKOP 20
PUDELKOP 30
PUDELKOP 40
END
```

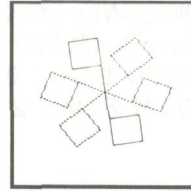


```
TO BRYLANTY
RT 45
REPEAT 4 [KWADRATY RT 90]
END
```



```
TO FLAGAP :ROZMIAR
FD :ROZMIAR
PUDELKOP :ROZMIAR
BK :ROZMIAR
END
```

```
TO 6 FLAG :ROZMIAR
REPEAT 6 [FLAGAP :ROZMIAR RT 60]
END
```



Jak się pewnie przekonałeś, możliwość kontrolowania rozmiarów rysowanych figur czyni grafikę bardziej użyteczną i interesującą.

DUŻE I MAŁE TRÓJKĄTY

Możesz także zdefiniować procedurę rysującą trójkąt, która wymaga parametru. Żądaj:

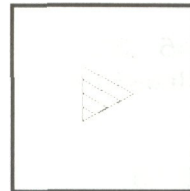
```
ED " TROJKAT
```

Teraz Edytor LOGO pokazuje Ci tekst procedury z kursorem ustawionym na TwTO:

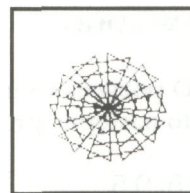
```
TO TROJKAT
REPEAT 3 [FD 30 RT 120]
END
```

Wypróbuj też takie procedury:

```
TO TROJKATY
TROJKATP 10
TROJKATP 20
TROJKATP 30
TROJKATP 40
END
```

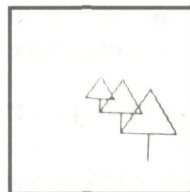


```
TO TROJGWIAZDA
REPEAT 10 [TROJKATY RT 36]
```



TO DRZEWO :BOK
RT 30 TROJKATP :BOK
RT 60 FD :BOK / 2
LT 90 BK :BOK / 2
END

TO DRZEWA
DRZEWO 30
DRZEWO 40
DRZEWO 50
END



ARYTMETYKA

W tym fragmencie dowiesz się jak używać operacji arytmetycznych w LOGO.

PR 5 + 3
LOGO odpowiada:
8

PR 4 * 23
LOGO odpowiada:
92

PR 345 - 32
LOGO odpowiada:
313

PR 25 / 5
LOGO odpowiada:
5

LICZBY W LOGO

LOGO pozwala Ci używać zarówno liczb całkowitych jak też ułamkowych. Oto następujące przykłady:

PR 2.5/0.5



5
PR25/6
4.16666666
PR 4*2.3
9.2
PR25/2
12.5

SŁOWNICZEK

W tym rozdziale poznałeś operacje LOGO:

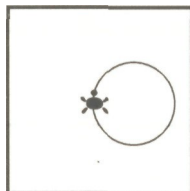
- / podzielić
- odjąć
- + dodać
- * pomnożyć
- : dwukropek

Żółw może rysować krzywe równie dobrze jak proste. Są one wykonywane poprzez wielokrotne wykonywanie małego kroku naprzód i małego skrętu.

Chcąc narysować okrąg, musisz żółwia obrócić o 360 stopni. Wypróbuj więc taki program:

```
REPEAT 360 [FD 1 RT 1 ]
```

Otrzymany okrąg wygląda doskonale, ale jego rysowanie zajęło mnóstwo czasu. Żółw musiał dwie Twoje instrukcje powtórzyć aż 360 razy - odpowiada to narysowaniu 90 kwadratów!



Jeśli chcesz swoje okręgi rysować szybciej, to musisz się zgodzić na pewien kompromis:

```
REPEAT 36 [FD 10 RT 10]
```

Teraz okrąg jest mniej regularny (tak naprawdę jest 36-ciokątem foremnym), ale za to żółw rysował go 10 razy szybciej. Rodzi się jednak pytanie, dlaczego chcieliśmy, abyś zmienił parametr dla FD z 1 na 10? Co by się stało, gdyby pozostała tam jedynka? Odpowiedź na to pytanie stanie się jasna po wykonaniu kilku eksperymentów. Najpierw napisz procedurę, która pozwoli Ci rysować okręgi różnych rozmiarów:

```
TO OKRAG :KROK  
REPEAT 36 [FD :KROK RT 10]  
END
```

Użyj jej teraz, wprowadzając różne liczby jako parametr:



OKRAG 1
OKRAG 5
OKRAG 10

Zauważ teraz, że wielkość okręgu jest proporcjonalna do parametru. Nie jest to zresztą zbyt dziwne - wszak każdy z okręgów zawiera tę samą liczbę kroków naprzód FD. W efekcie długość KROKu określa jednoznacznie obwód okręgu.

OKRĄG O ŻĄDANYM PROMIENIU

Często sytuacja wymaga rysowania okręgów o ściśle określonym promieniu. Chcąc wykorzystać procedurę OKRAG musisz każdorazowo policzyć KROK, który będzie odpowiadał żadanemu promieniowi. Naturalnie nie ma sensu, abyś liczył to sam - zlecisz to komputerowi. A oto sposób, w jaki można się do tego zabrać:

```
TO OKRAG.PROM :PROMIEN
OKRAG 2*3.14 *:PROMIEN /36
END
```

Liczba $2*3.14* :PROMIEN$ reprezentuje obwód okręgu ($2nR$). Obwód ten odpowiada 36 KROKom żółwia, a więc jeden musi mieć długość $2*3.14* :PROMIEN /36$.

A teraz pobaw się tymi nowymi procedurami - z pewnością wymyślisz dla nich więcej zastosowań, niż my mogliśmy Ci tu zaproponować.

ŻÓŁW RYSUJE ŁUKI

Wiele zastosowań wymaga rysowania nie pełnych okręgów, ale ich fragmentów, czyli łuków. Jest pewien sposób na rysowanie łuków, który możesz wykorzystać już teraz. Uruchom procedurę OKRAG i naciśnij BREAK - tak szybko, by żółw nie zdążył narysować całego okręgu. To, co otrzymasz będzie wprawdzie łukiem, ale niestety trudno Ci będzie w pewny sposób zdecydować o wielkości łuku - refleks jest jednak bardzo zawodny. Można to zresztą zrobić prościej i lepiej: trzeba tylko przerobić procedurę OKRAG tak, by miała ona teraz dwa parametry. Ten nowy będzie określał ile razy żółw ma wykonać mały krok naprzód i mały skręt w prawo.

Napisz:



EDIT "OKRAG

ATARI LOGO EDITOR udostępni Ci tekst procedury OKRAG:

```
TO OKRAG :KROK
REPEAT 36 [FD :KROK RT 10]
END
```

Łatwo sprawdzisz, że łuk 10 36 rysuje cały okrąg, ŁUK 10 18 jego połowę, ŁUK 10 9 ćwiartkę itd. Jednak jeszcze wygodniej byłoby, gdybyś mógł zamiast ilości kroków jako parametr traktować ilość stopni łuku, który ma zakreślić żółw. Nic prostszego!

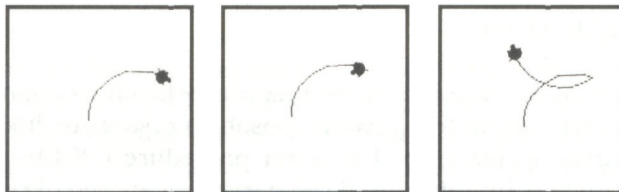
EDIT "LUK

przerób teraz tekst procedury żeby wyglądał tak:

```
TOLUK :KROK :STOPNIE
REPEAT :STOPNIE / 10 [FD :KROK RT 10]
END
```

A teraz narysuj rybę:

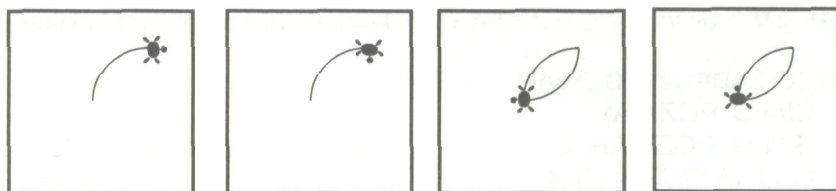
```
LUK 6 120
RT 120
LUK 6 120
```



Albo płatek kwiatu: oto procedura

```
TO PLATEK :ROZMIAR
LUK :ROZMIAR 90 RT 90
LUK : ROZMIAR 90 RT 90
END
```





i jej wykorzystanie:

```

PLATEK 3 PLATEK2
REPEAT 8 [PLATEK 3 PLATEK 2 RT 45]
REPEAT 4 [PLATEK 2 RT 45 PLATEK 3 RT 45]

```

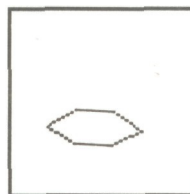
Zauważ, że linia w LOGO może być dłuższa, niż jedna linia na ekranie. LOGO sygnalizuje przeniesienie linii przez strzałkę "-", która pojawi się jako ostatnia w linii - reszta linii dopisywana jest niżej.

Podamy Ci przykład większego programu, który będzie rysował łabędzia. Używa on specjalnych procedur LEWYLUK, PRAWYLUK i LEWYOKRAG, których definicje znajdziesz w Dodatku A.

```

TO CIAŁO :ROZMIAR
RT 45
PLATEK :ROZMIAR
LT 45
END

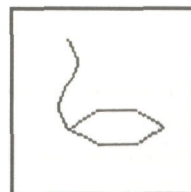
```



```

TO SZYJA :ROZMIAR
LT 45
PRAWYLUK :ROZMIAR 90
LEWYLUK :ROZMIAR 90
RT 45
END

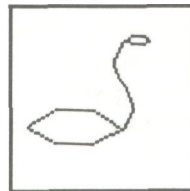
```



```

TO GŁOWA :ROZMIAR
LT 135
PLATEK :ROZMIAR
RT 135
END

```



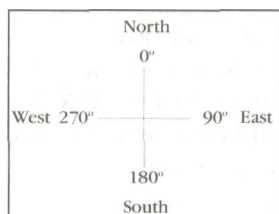
ŁABĘDŹ - procedura kontrolująca wykonanie programu wygląda tak:

```
TO LABEDZ :ROZMIAR  
CIALO :ROZMIAR  
SZYJA :ROZMIAR/2  
GLOWA :ROZMIAR/4  
END
```



"JAK WYGLĄDA POLE ŻÓŁWIA"

Żółw w każdym momencie ma pozycję i kierunek. Kierunek można opisać podając ilość stopni liczoną tak, jak w kompasie (tj. w kierunku ruchu wskazówek zegara) poczynając od zera (północ) umieszczonego na górze ekranu. Pokazuje to poniższy rysunek:



W pierwszym momencie jak również po każdym użyciu CS kierunek żółwia wynosi 0. Żądaniem meldunku o aktualnym kierunku żółwia jest specjalne słowo. Napisz kolejno:

CS
RT 90
PR READING

LOGO natychmiast odpowiada:

90

Tak więc HEADING podaje nam kierunek żółwia. Słowo HEADING należy do słownika LOGO. Różni się jednak od tych, które dotychczas poznałeś. Nie jest komendą, a operacją. Nie powoduje wykonania żądanej czynności, ale podaje pewien wynik (niekoniecznie liczbę), który może być użyty jako parametr w innej procedurze.

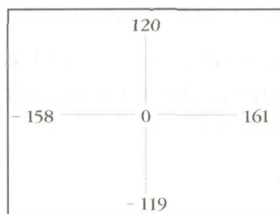
SETH (skrót od SET HEADING) jest komendą, nadającą żółwiowi kierunek, który podasz jako jej parametr. Istotna różnica pomiędzy SETH i RT czy LT polega na tym, że końcowy kierunek żółwia po wykonaniu komendy SETH nie zależy w żaden sposób od wcześniejszego. Obejrzyj to na następującym przykładzie:

SETH 90
RT 90
SETH 90

Pozycja żółwia jest opisywana przez dwie liczby, które pokazują jego położenie względem środka ekranu. W momencie startu POS (to skrót od POSITION) jest równe (0 0). Pierwsza z liczb określa położenie żółwia na poziomej osi, zwanej osią X. Jeżeli żółw znajduje się na zachód od centrum ekranu, to jest ona ujemna. Podobnie druga z nich reprezentuje położenie żółwia względem osi pionowej (oś Y) i jest ona ujemna, jeśli znajduje się on na południe od środka ekranu.

Pole żółwia można sobie wyobrazić jako siateczkę utworzoną ze współrzędnych. Współrzędne X są poziome, zaś Y - pionowe. Jeżeli żółw znajduje się w centrum ekranu to XCOR i YCOR są równe zero.

POS, XCOR, YCOR - to operacje języka LOGO. Początkowe wymiary ekranu są takie, jak na rysunku poniżej:



Wpisz teraz następujący przykład:

CS
LT 90
FD 30
PR POS

LOGO melduje:

- 30 0

A więc żółw znajduje się 30 kroków na wschód (w prawo) od środka w połowie wysokości ekranu.

Możesz również zażądać podania tylko jednej ze współrzędnych:

PR XCOR



Odpowiedź:

30

Próbuj dalej:

RT 90
FD 52
PR YCOR

Tym razem odpowiedź wynosi 52.

SETPOS (skrót od SET POSITION) przesuwa żółwia do pozycji, którą podamy jako parametr. Inaczej niż w przypadku FD i BK krańcowe położenie żółwia nie zależy od jego poprzedniej pozycji. HEADING żółwia pozostaje bez zmian.

Napisz:

SETPOS [50 -52] — umieść koniecznie spację pomiędzy 50 i -52!

W tym przykładzie 50 jest współrzędną X, zaś -52 współrzędną Y punktu, do którego przesunąłeś żółwia.

WRAP I WINDOW

Początek pracy żółwia odbywa się w trybie WRAP. Polega to na tym, że w momencie opuszczania ekranu z jednej strony, pojawia się on z drugiej i kontynuuje swój marsz zachowując kierunek. Jaśniejsze się to stanie po obejrzeniu na ekranie takiego przykładu:

CS
FD 500
PR POS

Odpowiedzią LOGO będzie 0 20, a nie 500.

Komenda WINDOW powoduje zmianę tego stanu rzeczy - odtąd żółw nie będzie już wracał z drugiej strony ekranu. Dlatego też często będzie się zdarzać, że nie będziesz go wcale widział na ekranie. Nie mniej - zawsze wykona wszystkie polecenia, które mu wydasz.

WINDOW



FD 500
PR POS

Tym razem odpowiedź będzie 0 500. Jeżeli stracisz kontrolę nad żółwem, zawsze możesz go przywołać komendą CS. Powrót do trybu WRAP odbywa się poprzez wydanie komendy WRAP. WRAP i WINDOW powodują w pierwszym momencie wyczyszczenie ekranu i ustawienie żółwia w pozycji wyjściowej.

UŻYCIE OPERACJI POS W RYSOWANIU

Jest to bardzo łatwa i prosta droga do narysowania trójkąta prostokątnego o zadanych przyprostokątnych. Zaczynij od zapamiętania początkowej pozycji żółwia. Naturalnie nie chodzi o to, byś sam tę pozycję zapamiętał, lecz by to zrobił komputer. Służy do tego nowa dla Ciebie komenda MAKE:

CS
MAKE "START POS

Komenda ta wykonuje dwie czynności: umieszcza wartość operacji POS w przestrzeni roboczej : nadaje jej nazwę START. Jeżeli więc zaży-
czysz sobie od LOGO:

PR :START

a żółw był w tym momencie w środku ekranu, to otrzymasz odpowiedź:

00

Wróćmy do trójkąta: najpierw narysuj obie przyprostokątne

FD 33
RT 90
FD 42

Teraz przyszedł czas na użycie komendy SETPOS:

SETPOS :START

Żółw zostanie przesunięty do pozycji :START, a że pióro miał

opuszczone, więc zostawi ślad, czyli przeciwprostokątną Twojego trójkąta. Niżej znajdziesz procedurę rysującą trójkąt prostokątny o przyprostokątnych długości BOK 1 i BOK 2:

```
TOTROJKAT :BOK1 :BOK2
MAKE "START POS
FD:BOK1
RT 90
FD :BOK 2
SETPOS :START
END
```

Wypróbuj działanie:

```
CS
TROJKAT 40 50
SETH 0
TROJKAT 75 20
```

SŁOWNICZEK

W tym rozdziale zapoznałeś się z następującymi słowami kluczowymi
LOGO:

Komendy:

```
SETH
SETPOS
WINDOW
WRAP
```

Operacje:

```
HEADING
POS
XCOR
YCOR
```

ROZDZIAŁ XV

"W KRAINIE WIELOKĄTÓW I SPIRALI"

Umiesz już w tej chwili kontrolować ilość kroków, które wykonuje żółw, a także kąt o jaki skręca. Zdziwiające jest, jak wiele różnych ciekawych rysunków możesz wykonać zmieniając tylko dwie litery.

Poniższa procedura wymaga dwóch parametrów: pierwszy określa odległość pokonywaną przez żółwia, a drugi kąt obrotu.

```
TO WIELOBOK :BOK :KAT
FD :BOK
RT :KAT
WIELOBOK : BOK : KAT
END
```

A teraz uruchom procedurę:

```
WIELOBOK 30 90
```

Żółw nie zatrzymuje się, gdyż wielobok ciągle każe mu wykonać następny obrót i tak bez końca. Musisz to przerwać naciskając klawisz BREAK.

LOGO reaguje meldunkiem:

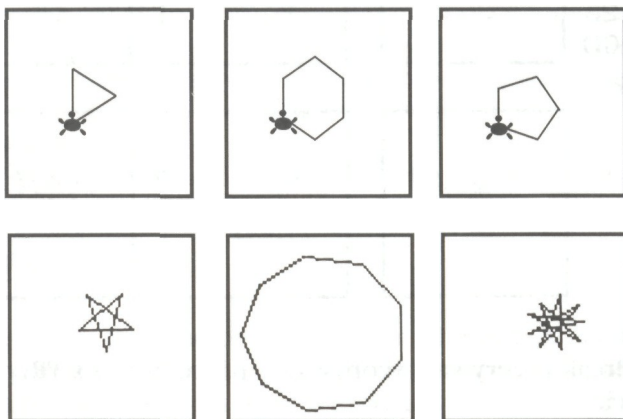
```
STOPPED! IN WIELOBOK
```

Pokazujemy Ci tutaj kilka przykładów kształtów uzyskanych procedurą WIELOBOK przy różnych parametrach:

```
WIELOBOK 30 120
WIELOBOK 30 60
WIELOBOK 30 72
WIELOBOK 30 144
WIELOBOK 30 40
WIELOBOK 30 160
```



WIELOBOK jest procedurą rekurencyjną. Oznacza to, że WIELOBOK sam się wywołuje w trakcie wykonywania. Każda procedura w LOGO ma swoją własną nazwę i jest w pełni samodzielna. Może zawierać instrukcje wywoływania innych procedur. W szczególności, jeżeli wywołuje samą siebie, to jest nazywana procedurą rekurencyjną.



SPIRALE

Procedura WIELOBOK każe żółtowi rysować figury zamknięte. Żółw chodząc w przód, obracając się wraca w miejsce startu. W szczególności gdy obraca się o 0° lub 360° (lub wielokrotność 360), rysuje linię prostą.

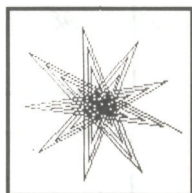
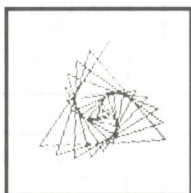
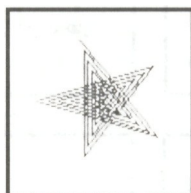
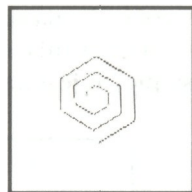
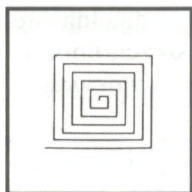
Rysując spiralę żółw nie może wracać w miejsce startu. Powinien zwiększać długość swojego kroku przy każdym obiegu ustalonego punktu obranego przez nas jako środek spirali. W ten sposób będzie się oddalać coraz bardziej i bardziej. Możesz zmusić go do tego dodając każdorazowo trochę do :BOK w linii rekurencyjnej procedury WIELOBOK.

W ten sposób przez niewielką modyfikację procedury WIELOBOK możesz otrzymać procedurę rysującą spiralę:

```
TO SPIRALA :BOK :KAT
FD :BOK
RT :KAT
SPIRALA :BOK + 6 :KAT
END
```



SPIRALA 5 90
 SPIRALA 5 120
 SPIRALA 5 60
 SPIRALA 5 144
 SPIRALA 5 125
 SPIRALA 5 160



Pamiętaj, że Break przerywa wykonywanie procedury, a CTRL F odsłania cały rysunek.

Możesz tę procedurę zmienić jeszcze bardziej, dodając jeszcze jeden parametr, który będzie dodawany do :BOKu zamiast szóstki.

W ten sposób możesz regulować gęstość rysowanej spirali. Nowa definicja powinna wyglądać tak:

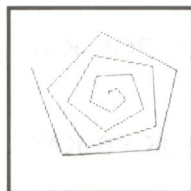
```

TO SPIRALA :BOK :KAT :SKOK
FD :BOK
RT :KAT
SPIRALA :BOK + :SKOK :KAT :SKOK
END
  
```

Obejrzyj i porównaj rysunki wykonywane na rozkaz:

```

SPIRALA 5 75 1
SPIRALA 5 75 2
  
```



ROZDZIAŁ XVI

"SPECJALNE MOŻLIWOŚCI GRAFIKI ŻÓŁWIA W ATARI LOGO"

Grafika żółwia w LOGO przeznaczona dla komputera ATARI charakteryzuje się szczególnie dużymi możliwościami. Pozwala ona użyć jednocześnie aż czterech żółwi. Ten którym posługiwałeś się dotychczas jest tylko jednym z nich. W dodatku wszystkie cztery mogą tworzyć rysunki jednocześnie.

W tym rozdziale nauczysz się jak dysponować wszystkimi żółwiami naraz oraz każdym z osobna.

CZTERY ŻÓŁWIE

Żółwie są ponumerowane od 0 do 3. Poprzez komendę TELL możesz sprecyzować, któremu żółwiowi pragniesz wydawać rozkazy.

Na przykład, żeby ustawić w szeregu cztery żółwie podaj następujące instrukcje:

```
ST  
FD 40
```

W momencie startu LOGO traktuje wszystkie Twoje komendy tak, jakby były przeznaczone dla żółwia o numerze 0. Ten stan rzeczy trwa tak długo, jak długo nie wydasz innego polecenia.

```
TELL 1
```

Żółw z numerem 1 pojawia się na środku ekranu.

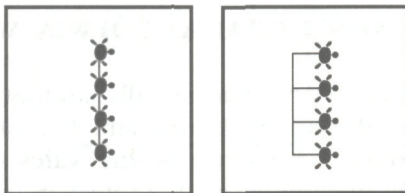
Jeśli żółw nie pojawi się po komendzie TELL spróbuj wywołać go komendąST:

```
FD 20  
TELL 2  
BK 20  
TELL 3
```

Chcąc adresować komendy jednocześnie do kilku żółwi musisz umieścić ich numery w nawiasie kwadratowym.



TELL [0 1 2 3]
RT 90
FD 20



Wszystkie żółwie robią "w prawo zwrot" i 20 kroków na przód.

ŻÓŁWIE W RUCHU

Komendą SETSP możesz wprawić żółwie w ruch nadając im stałą prędkość.

SETSP 20

Ten przykład pokazuje jak nadać wszystkim żółwiom prędkość 20. Większa liczba wprowadzona jako parametr spowoduje szybsze poruszanie się żółwi. Poeksperymentuj zmieniając wartość prędkości:

SETSP 50
SETSP 10

Żółwie pozostawiają za sobą ślad, gdyż ich pióra są opuszczone. Możesz jednak wymazać te linie poprzez PE (od ang. pen eraser).

Gdy wprawisz żółwie w ruch będą się poruszać zachowując kierunek, który miały w momencie startu. O ile zmienimy żółwiowi kierunek w trakcie ruchu, wykona on faktycznie skręt.

Spróbuj zrobić ten eksperyment z żółwiem 1:

TELL 1
RT 180

Zauważyłeś pewnie, że LOGO jednocześnie poruszając żółwiami jest gotowy wykonywać inne Twoje rozkazy. Jest to wyraźne odróżnienie od Grafiki Żółwia, z którą mieliśmy do czynienia dotychczas.

Jeżeli znudziły Ci się cztery żółwie możesz wrócić do pierwotnego

stanu - z jednym żółciem w następujący sposób:

```
TELL [0 1 2 3]  
CS
```

Wszystkie żółcie są umieszczone w centrum ekranu jeden na drugim. Można się też ich pozbyć komendą:

```
HT
```

i z powrotem wywołać żółcia 0:

```
TELL 0 ST
```

Żeby wrócić do stanu pierwotnego opuść jeszcze na dół pióro:

```
PD
```

KOLOR ŻÓŁWIA

Jak się zapewne zorientowałeś, każdy z żółci w momencie pierwszego pojawienia się na ekranie był innego koloru. Nie są one jednak stałe i możesz je w każdej chwili zmienić stosując komendę SETC (od SET COLOR). Na przykład polecenia:

```
TELL 0  
SETC 20
```

powodują zmianę koloru żółcia 0 na złotą. Jak się pewnie domyślasz kolory są ponumerowane, a ich numery odpowiadają dokładnie tym, które podaliśmy w tabelce w Rozdziale 7.

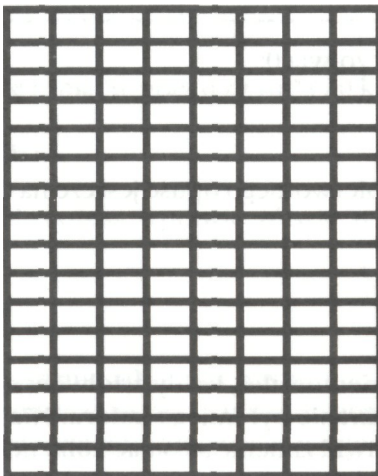
KSZTAŁT ŻÓŁWIA

Jeżeli zwykły kształt żółcia znudzi Ci się, możesz go zmienić na inny, wedle Twego uznania - może to być kształt rakiety, gwiazdki samochodu wyścigowego lub jakikolwiek inny - jedynym limitem jest Twoja wyobraźnia. Do tych zmian służy Edytor Kształtu, a komputer może w swej pamięci przechowywać jednocześnie piętnaście różnych wersji wyglądu żółcia. Gdy się zdecydujesz na taką zmianę wydaj komendę EDSH (od EDit SHape) wraz z parametrem, którym powinna być liczba od 1 do 15 -

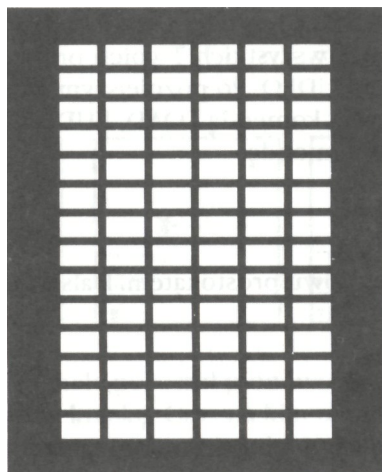
- pod takim numerem będzie w pamięci przechowywany kształt, który zaraz określisz.

Na ekranie ujrzysz siatkę złożoną z ośmiu kolumn po szesnaście prostokątów w każdej. Kursor będzie się znajdował w lewym górnym rogu. Możesz go zupełnie dowolnie poruszać używając kombinacji klawiszy CTRL→, CTRL←, CTRL↑, CTRL↓

EDSH 1



W pierwszym momencie wszystkie prostokąty będą puste. Ty możesz je wypełnić. Robi się to bardzo prosto - wystarczy ustawić kursor na wybranym prostokącie i nacisnąć klawisz spacji, a wypełni się on. Jeżeli naciśniesz ten klawisz jeszcze raz, to prostokąt znowu stanie się pusty. Stosując tę prostą technikę możesz namalować na siatce dowolny rysunek - stanie się on nową postacią "żółwia". Na początek proponujemy Ci nieskomplikowany kształt prostokąta, który widzisz poniżej.



Po wykonaniu go naciśnij klawisz ESC, żeby wyjść z Edytora Kształtu. Wykorzystaj teraz komendę SETSH (od SET SHape) aby uczynić żółwia 0 tym prostokątem:

```
CS
TELL 0
ST
SETSH 1
```

Żółw w swej nowej postaci zachowa wszystkie swoje poprzednie właściwości z jednym małym wyjątkiem: odtąd nie będzie sygnalizował obrotami swej figurki kierunku, w którym ma się poruszać.

PRZECHOWYWANIE POSTACI ŻÓŁWIA

Od momentu opuszczenia Edytora Kształtu zdefiniowany przez Ciebie rysunek jest przechowywany w przestrzeni roboczej komputera. Nie jest to jednak przechowywanie wieczne - każde wyłączenie komputera opróżnia całą jej zawartość, więc i Twoje kształty znikną. Oczywiście i na to jest rada: jest nią operacja GETSH (GET SHape).

Oto przykład jej wykorzystania:

```
MAKE " PUDLO GETSH 1
```



Odtąd komputer przechowuje kształt prostokąta zdefiniowany jako kształt 1 pod nazwą PUDLO. Jeżeli teraz wydasz komendę SAVE, to na taśmę magnetyczną oprócz wszystkich Twoich procedur przepisany zostanie także kształt 1 jako PUDLO. Po późniejszym wczytaniu zbioru z kasety do pamięci komputera komendą LOAD, PUDLO trafi do niej z powrotem. Gdy teraz wydasz komendę:

```
PUTSH1:PUDLO
```

Kształt 1 stanie się znowu prostokątem. Dalszymi komendami:

```
TELL 0  
SETSH 1
```

nadasz "żółwiowi" postać prostokąta.

WYKRYWANIE ZDERZEŃ

ATARI LOGO daje Ci możliwość zdalnego wykrywania kolizji pomiędzy dwoma żółwiami oraz żółwiem a rysunkiem znajdującym się na ekranie. Możliwe przypadki kolizji są zakodowane jako liczby od 0 do 21, określające które z żółwi się zderzyły, bądź który żółw najechał na który rysunek, przy czym LOGO rozróżnia trzy rodzaje rysunków, w zależności od pióra (0, 1 lub 2) którym zostały narysowane. Dokładny opis kolizji określonych poszczególnymi liczbami znajdziesz w Dodatku B.

Liczb tych używa się jako parametrów komendy WHEN.

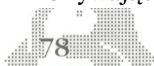
Najprościej będzie, jeśli przyjrzesz się jej działaniu na przykładzie:

```
TELL [0 1 2 3]  
CS  
SETSH 0  
HT
```

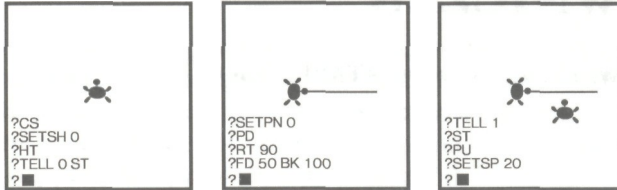
Teraz ekran jest już zupełnie czysty.

```
TELL 0 ST  
SETPN 0  
PD  
RT 90  
FD 50 BK 100
```

Używając żółwia i pióra 0 narysowałeś poziomą linię:



TELL 1
ST
PU
SETSP 20



Teraz żółw będzie się stale poruszał pionowo z prędkością 20, co jakiś czas przecinając pionową kreskę. Czas na kluczową komendę. Po jej wydaniu żółw po każdorazowym najechaniu na kreskę będzie zmieniał kierunek marszu o 180 stopni.

WHEN 4 [RT 180]

Liczba 4 oznacza kolizję pomiędzy żółwiem nr 1 i rysunkiem wykonanym piórem 0. Liczbę tę (jak również pozostałe) można odtworzyć stosując słowa kluczowe OVER i TOUCHING. O ich użyciu dowiesz się więcej, jeżeli przeczytasz Dodatek B.

Komenda WHEN przywołuje Ci na pomoc "Stworka WHEN", żyjącego w świecie ATARI LOGO. Cały swój czas spędza on na oczekiwaniu na jakąkolwiek kolizję na ekranie. Gdy tylko ona zajdzie, rozkazuje on żółwiowi, którym właśnie się posługujesz, wykonać wszystkie polecenia, które otrzymał od Ciebie w liście. Natychmiast potem powraca do obserwowania ekranu. Będzie to robił dopóki nie użyjesz komendy CS lub nie pozwolisz mu odpocząć, co można zrobić komendą:

WHEN 4 []

Zauważ, że polecenia stworka WHEN wykonuje ten żółw, którego Ty w tym momencie używasz, a wcale nie musi to być ten, którego ruch śledzi WHEN. Aby obejrzeć ten efekt wydaj komendę:

TELL 0

Odtąd to żółw 0 będzie robił "w tył zwrot", gdy żółw 1 będzie

nadeptywał na kreskę.

SŁOWNICZEK

A oto nowe słowa kluczowe ATARI LOGO:

Komendy:

EDSH

PUTSH

SETC

SETSH

SETSP

TELL

WHEN

Operacje:

GETSH



ROZDZIAŁ XVII

"PROJEKTOWANIE GIER"

Rozdział ten wyjaśnia jak projektować gry używając dynamiki żółwia. Nasza propozycja - to prosta gra nawigacyjna. Tarcza, kilka przeszkód i pojawiający się na ekranie żółw są naszymi rekwizytami. Zadaniem gracza jest przeprowadzenie żółwia w możliwie najmniejszej liczbie kroków do tarczy, oczywiście z pominięciem przeszkód.

Tarczą może być dowolny "Żółwiowy" rysunek, bądź też inny żółw. W naszym przykładzie będziemy jako tarczę traktować żółwia 0, jako nawigatora żółwia 1, a jako czujnik wykrywający kolizję między nimi stworzka WHEN.

Tego rodzaju gry stają się prostsze i łatwiejsze gdy realizuje się je etapami. Dlatego dla uproszczenia stworzymy pierwszą wersję gry bez przeszkód. Żółw 1 będzie w ruchu, a gracz będzie sterować nim w kierunku tarczy standardowymi komendami LOGO RT 45 lub LT 30. W drugim etapie postaramy się o większe wyrafinowanie i oznaczymy specjalne klawisze do kierowania żółwiem. I wreszcie na końcu dostawimy tor przeszkód.

STRUKTURA GRY

Na początku musimy ustawić tarczę, a potem będziemy potrzebować żółwia. Aby zapewnić sobie te rekwizyty wystarczy jedna procedura na przykład taka, jaką proponujemy poniżej.

Procedura REKWIZYTY ustawia żółwia w losowo wybranym miejscu ekranu nadając mu kierunek 0 (pyszczkiem na północ):

```
TO REKWIZYTY
PENUP HOME
RT RANDOM 360
FD RANDOM 80
SETH 0
END
```

Rezultatem operacji RANDOM jest wylosowana przez LOGO liczba. Losowanie odbywa się zawsze spośród liczb mniejszych od wybranej przez nas i wprowadzonej do komputera jako parametr operacji



RANDOM. W procedurze REKWIZYTY żółt obraca się w prawo o kąt, który może wynosić od 0 do 360. Liczba losowa wyliczana jest na nowo za każdym użyciem RANDOM. Parametr instrukcji FD także jest liczbą losową tym razem nie większą od 79. Ostatnia instrukcja procedury ustawia żółwia pyszczkiem na północ. Procedurę REKWIZYTY oprócz tarczy i żółwia można różnie dobrze ustawić przeszkody.

Niżej przedstawiona przez nas procedura PLANSZA korzysta dwukrotnie z procedury REKWIZYTY: raz w celu ustawienia losowo żółwia - tarczy drugi raz żółwia - nawigatora.

```
TO PLANSZA
CS
TELL 0 ST
REKWIZYTY
TELL 1 ST
REKWIZYTY
END
```

Wypróbuj teraz tę procedurę. Gra stanie się bardziej przejrzysta jeśli trzeba będzie się w jakiś sposób odróżnić od nawigatora. Dlatego zastąpmy klasyczny kształt żółwia służącego nam za tarczę kształtem pustego prostokąta.

Zdefiniowany on został w poprzednim rozdziale jako Kształt 1. Dołączymy więc instrukcję SETSH 1 do procedury PLANSZA. Upewnij się, że kształt SETSH 1 jest określony.

```
TO PLANSZA
CS
TELL 0 ST
REKWIZYTY
SETSH 1
TELL 1 ST
REKWIZYTY
END
```

teraz wykorzystaj tę planszę do procedury GRA:

```
TO GRA
PLANSZA
SETSP 30
WHEN 19 [SETSP 0]
END
```



Ostatnia instrukcja zatrzymuje żółwia w chwili osiągnięcia celu - tarczy. Numer 19 oznacza kolizję między żółwiem 0 (tarcza) i 1 (nawigator). Wypróbujmy grę kilka razy:

GRA
RT 45
LT 10

PRZYSTOSOWANIE KLAWISZY DO STEROWANIA GRA

Uruchamiając swoje programy można stosować wiele technicznych rozwiązań ułatwiających nam współpracę z komputerem. Można zadawać LOGO pytania i oczekiwać na nie odpowiedzi słowami bądź pełnymi zdaniem. Można też, jak w przypadku naszego programu "rozmawiać" z komputerem przez naciskanie klawiszy. Konieczna jest do tego operacja RC.

Napisz:

PR RC

W tej chwili LOGO czeka na wprowadzenie znaku z klawiatury. Naciśnij A. RC "odbiera" znaczenie klawisza A i przyporządkowuje je komendzie PRINT. Efektem tego jest wypisywanie A na ekranie.

LOGO nie czeka na następne napisy, działa bezpośrednio po naciśnięciu A.

Poeksperymentuj z operacją RC. Zauważ, że pisząc samo RC (wprowadzone klawiszem RETURN) i następnie wprowadzając znak z klawiatury na przykład X, nie będziesz rozumiany przez LOGO.

Da Ci o tym znać napis:

YOU DON'T SAY WHAT TO DO WITH X

RC jest operacją o podobnym działaniu jak HEADING i POS. Może być użyta jako parametr innych komend i operacji. Na przykład nazwijmy rezultat operacji RC przy pomocy MAKE:

MAKE "KLAWISZ RC

Napisz teraz znak. (Zauważ, że nie pojawi się on na ekranie, inaczej mówiąc LOGO nie daje "echa", gdy coś wprowadzasz). Teraz możesz

mówić o tym klawiszu nazywając go :KLAWISZ

```
PRINT :KLAWISZ
```

LOGO odpowie:

```
Z
```

(albo inną literą, zawsze jednak taką, którą ty naciśniesz).

Powyższy pomysł wykorzystamy w następniej procedurze. Wyobraźmy sobie najpierw jej działanie. Niech naciśnięcie klawisza L powoduje obrót o 15 stopni w lewo, zaś P taki sam obrót w prawo. Nazwijmy procedurę STEROWANIE:

```
TO STEROWANIE
MAKE "ODPOWIEDZ RC
IF :ODPOWIEDZ = "P [RT 15]
IF :ODPOWIEDZ = "L [LT 15]
END
```

ODPOWIEDZ jest nazwą jaką nadaliśmy rezultatowi operacji RC. Procedura ta sprawdza ODPOWIEDZ korzystając ze słowa kluczowego IF. Instrukcja ta wymaga dwóch parametrów: pierwszym musi być warunek, któremu w danej chwili odpowiada prawda (TRUE), bądź fałsz (FALSE), drugim lista rozkazów. W naszym przykładzie sprawdzamy warunek przy pomocy operacji = (równość). Wymaga ona dwóch parametrów i daje w wyniku prawdę (TRUE) jeżeli oba są takie same i fałsz (FALSE) w przeciwnym wypadku.

LOGO powinno uruchamiać procedurę STEROWANIE, gdy wprowadzisz coś z klawiatury. Zrobisz to łatwo przy pomocy operacji KEYP. Daje ona w wyniku prawdę, gdy klawisz naciśniemy i fałsz jeśli nie. Możesz więc napisać procedurę uruchamiając STEROWANIE, wtedy i tylko wtedy gdy klawisz zostanie naciśnięty.

Napisz:

```
TO ZABAWA
IF KEYP [STEROWANIE]
ZABAWA
END
```

Zauważ, że ZABAWA jest rekurencyjna. Ostatnia linia tej procedury



wywołuje ją na nowo. Zatrzymać ją możesz klawiszem BREAK.
wypróbuj:

```
GRA
ZABAWA
```

Naciskaj L i P żeby kierować żółciem. Jeśli nic się nie dzieje przy wciskaniu L i P sprawdź czy przypadkiem nie został wciśnięty klawisz CAPSLOWR. Jeśli tak LOGO nie rozumie Twoich rozkazów. Musisz wrócić do pisania dużymi literami naciskając SHIFT i CAPS LOWR jednocześnie.

ROZSZERZENIE PROJEKTU GRY

Teraz na bazie procedur GRA, ZABAWA i PLANSZA skonstruujemy lepszy projekt. Najpierw w procedurze PLANSZA dodamy kilka żółtowych rysunków, które posłużą nam jako przeszkody:

```
TO PLANSZA
CS
TELL 0 ST
REPEAT 3 [REKWIZYTY PD FD 20]
REKWIZYTY
SETSH 1
TELL 1 ST
REKWIZYTY
END
```

Trzecia komenda przy pomocy instrukcji REPEAT służy do narysowania trzech krótkich odcinków w losowych miejscach ekranu.

GRA powinna dawać znać, kiedy stworek WHEN wykryje zderzenie żółcia z przeszkodą. Numer kolizji między żółciem 1 i piórem 0 jest 4. Gdy taka kolizja nastąpi żółt powinien cofnąć się o kilka kroków. Eleganckim rozwiązaniem będzie umieszczenie reguł zabawy na początku procedury GRA:

```
TO GRA
REGULY
PLANSZA
SETSP 10
WHEN 19 [SETSP 0]
```



```

WHEN 4 [BK 6]
ZABAWA
END
TOREGULY
SS
PRINT [DOPROWADZ ZOLWIA DO TARCZY]
PRINT [STERUJ KLAWISZAMI P i L]
END

```

Wypróbuj GRE. Napisz:

```
GRA
```

Jest bardzo dobra, ale może być jeszcze lepsza. Teraz można przerwać tylko naciskając BREAK. Przerób ją tak, żeby zatrzymywała się po skończeniu. Ale skąd ma się LOGO dowiedzieć, że gra się skończyła? Jedną z metod jest śledzenie prędkości żółwia. Zauważ, że stworek WHEN, gdy tylko dwa żółwie się spotkają umieszcza w procedurze GRA rozkaz SETSP 0. Zabawa musi więc zawierać sprawdzenie czy prędkość jest = 0. Wtedy gra się kończy i ZABAWA musi się zatrzymać.

```

TO ZABAWA
IF SPEED = 0 [STOP]
IF KEYP [STEROWANIE]
ZABAWA
END

```

SPEED jest operacją, której rezultatem jest wartość prędkości, za którą w danej chwili porusza się żółw. Teraz cała gra jest gotowa. Spróbuj więc zagrać:

```
GRA
```

Pamiętaj, że sterujesz żółwiami komendami P i L. Gdy żółw osiągnie tarczę procedura się zatrzyma i LOGO wypisze znak zachęty na ekranie.

Jesteś w stanie zaadaptować tę grę i techniki w niej zawarte do wykonania innych gier podobnego typu. Możesz także udoskonalić naszą grę. Na przykład możesz zastąpić żółwia 0 innym kształtem, albo zlecić LOGO zapisywanie Twoich wyników. Możesz także zmienić reguły gry, tak żeby gracz tracił punkt przy każdym zderzeniu z przeszkodą.



SŁOWNICZEK

W tej części poznaliśmy komendy:

IF

STOP

oraz operacje:

KEYP

RANDOM

RC

SPEED

ROZDZIAŁ XVIII

"PROCEDURY REKURENCYJNE"

Jedną z cenniejszych cech LOGO jest możliwość pisania programów składających się z odrębnych procedur. Każda z nich ma swoją własną nazwę i stanowi samodzielną całość. Procedury mogą wywoływać (używać) inne procedury, także dopuszczalne jest by procedura sama siebie wywoływała. W takim przypadku nazywamy ją procedurą rekurencyjną. Posługiwaliśmy się już takimi procedurami w rozdziale 15 (Na przykład SPIRALA i WIELOBOK).

ZATRZYMANIE PROCEDUR REKURENCYJNYCH

Można podać wiele przykładów operacji, które umieszczone wewnątrz procedury spowodują jej zatrzymanie. Poniżej znajdziesz prosty przykład wykorzystujący operację KEYP:

```
TO SPIRALA :BOK :KAT :SKOK
IFKEYP[STOP]
FD:BOK
RT:KAT
SPIRALA:BOK+:SKOK:KAT:SKOK
END
```

Wynikiem KEYP jest TRUE (prawda) ilekroć jakkolwiek klawisz jest naciśnięty. Użycie tej operacji wraz z IF pozwala Ci zatrzymać procedurę naciśnięciem dowolnego klawisza. Ten sposób wymaga ingerencji człowieka. Teraz zobaczysz przykład, który się odbywa bez tego. Jeżeli zmienisz pierwszą linię SPIRALA na:

```
IF:BOK> 150 [STOP]
```

to zatrzymanie nastąpi w momencie, gdy :BOK stanie się większy niż 150.

Reguła zatrzymania WIELOBOK może być trochę sprytniejsza. Program ten rysuje figurę w ten sposób, że żółw wraca do swojego pierwotnego stanu. Co za tym idzie, musi wykonać obrót o kąt pełny czyli 360°. Jednak czasami obrót jest wielokrotnością 360-ciu stopni.



Na szczęście potrzebujesz tylko informacji kiedy kierunek żółwia będzie taki sam, jak na początku. Dokonaj więc w WIELOBOK zmiany. Operacja zatrzymania będzie wyglądała tak:

```
IF HEADING = :START [STOP]
```

Umieść ją w procedurze po komendzie RT - gdybyś to zrobił wcześniej, reguła zatrzyma pracę zanim się ona na dobrze zacznie.

```
TO WIELOBOK :BOK :KAT
FD:BOK
RT :KAT
WIELOBOK :BOK :KAT
END
```

```
TO SPIRALA :BOK :KAT :SKOK
FD:BOK
RT :KAT
SPIRALA:BOK + :SKOK:KAT:SKOK
END
```

WIELOBOK wywołuje WIELOBOK w części definicyjnej i podobnie SPIRALA wywołuje SPIRALA.

Zastanówmy się nad tym procesem. Wyobraźmy sobie, że LOGO ma nieograniczone zasoby pomocników, będących stworkami żyjącymi w komputerze. Przy każdym wywołaniu procedury "wzywamy" pomocnika, żeby śledził jego definicję. Krasnoludek zaczyna wykonywać instrukcje. Robi to ściągając na pomoc inne krasnoludki. Najczęściej kilku pomocników potrzeba do zrealizowania jednej procedury.

Na przykład, gdy wywołujemy WIELOBOK wzywany jest krasnoludek FD. Kiedy skończymy pracę wołamy krasnoludka RT. I tak dalej coraz to nowe krasnoludki. Drugi wielobok wzywa krasnoludków FD i RT i trzeciego krasnoludka WIELOBOK. W tym czasie pierwszy pomocnik WIELOBOK czeka na skończenie pracy przez drugiego. W tym czasie kończą pracę krasnoludki FD i RT (nie jest istotne, kto wzywał ich na pomoc) krasnoludek WIELOBOK nigdy nie skończy swojej roboty; będzie ciągle wzywać nowych pomocników WIELOBOK.

Proces WIELOBOK jest kontynuowany, aż do naciśnięcia BREAK. Nie wszystkie procedury rekurencyjne działają w ten sposób. Przy spełnieniu pewnych warunków mogą się samoczynnie zatrzymywać.



```
RT :KAT
IF HEADING = :START [STOP]
WIELOBOK :BOK :KAT
END
```

Teraz przed każdorazowym wywołaniem nie zapomnij poinformować komputera o pozycji początkowej.

```
MAKE "START HEADING
WIELOBOK 10 36
```

Teraz już komputer będzie wiedział kiedy zakończyć rysowanie. Nie jest to najlepsze rozwiązanie - jeśli zapomnisz LOGO poinformować o pozycji - WIELOBOK nie będzie działał. Najlepiej będzie, jeśli ten obowiązek zwalisz na samą procedurę. Program - który napiszesz będzie się składał z dwóch procedur. WIELOBOK 1 jest prostą przeróbką WIELOBOK.

```
TO WIELOBOK 1 :BOK :KAT :START
FD :BOK
RT :KAT
IF HEADING = :START [STOP]
WIELOBOK 1 :BOK :KAT :START
END
```

Teraz napisz jeszcze nowy WIELOBOK, który będzie zapamiętywał START i podawał jego wartość do WIELOBOK 1.

```
TO WIELOBOK :BOK :KAT
WIELOBOK :BOK :KAT HEADING
END
```

Ta procedura jest już w pełni zadowalająca. Zwróć uwagę, że HEADING jest operacją i dlatego przed tym słowem nie stawiamy: (dwukropki), tak jak to czynimy, gdy chcemy odczytać wartość jakiejś zmiennej.

Łącząc elementy procedur rysujących wieloboki i spirale możesz otrzymać nowe programy tworzące obrazy.

W tym miejscu nasza rola się kończy: wiesz już dostatecznie dużo, aby dalej poznawać grafikę żółwia ATARI LOGO samemu.

Przyjemnej zabawy!



DODATEK A

UŻYTECZNE NARZĘDZIA: PROCEDURY RYSUJĄCE ŁUKI I OKRĘGI

```
TO PRAWYLUK :PROMIEN :STOPNIE
PRLUK 1 .174532* :PROMIEN :STOPNIE /10
IF 0 = REMAINDER :STOPNIE 10 [STOP]
FD .174532* :PROMIEN ( 10* REMAINDER :STOPNIE 10
RT REMAINDER :STOPNIE 10
END
```

```
TO PRAWYLUK :PROMIEN :STOPNIE
PRAWYLUK X :PROMIEN :STOPNIE
END
```

```
TO LEWYLUK :PROMIEN :STOPNIE
LWLUK1 .174532* :PROMIEN :STOPNIE /10
IF 0 =REMAINDER :STOPNIE 10 [STOP]
FD .174532* :PROMIEN /10 X REMAINDER :STOPNIE
LT REMAINDER :STOPNIE 10
END
```

```
TO LWLUK :PROMIEN :STOPNIE
LEWYLUK :PROMIEN :STOPNIE
END
```

```
TO PRLUK1 :KROK :ILOSC
REPEAT :ILOŚĆ [RT 5 FD :KROK RT5]
END
```

```
TO LEWYOKRAG :PROMIEN
LWLUK1 .174532* :PROMIEN 36
END
```

```
TOPRAWYOKRAG :PROMIEN
PRLUK1 .174532* :PROMIEN
END
```

Uwaga: Procedury w rzeczywistości rysują fragmenty trzydziestosześcioramiennego foremnego ($174532 = 2 * n / 36$).



DODATEK B

ATARI LOGO posiada możliwość wykrywania kolizji na ekranie. Dzieje się to za pomocą komendy WHEN, która przywołuje stworka WHEN oczekującego na określony rodzaj kolizji, aby w razie jej zajścia spowodować wykonanie serii komend. Patrz - Rozdział 16. Istnieje możliwość użycia operacji OVER i TOUCHING jako pierwszego parametru komendy WHEN. OVER posiada dwa parametry: numer żółwia i numer pióra. Na przykład komenda WHEN OVER 1 0 [RT 180] jest identyczna z rozpatrywaną w Rozdziale 16 komendą WHEN 4 [RT 180].

Operacja TOUCHING działa podobnie, ale jej parametrami są dwa numery żółwi. Przykładowo komenda WHEN TOUCHING 1 2 [RT 180] jest równoznaczna z komendą WHEN 21 [RT 180].

W umieszczonej niżej tabeli umieszczone są liczby opisujące poszczególne rodzaje kolizji. Numery 0-14 (za wyjątkiem 3,7 i 11) opisują kolizje pomiędzy żółwiami... 3,7 i 15 są użyte do kodowania specjalnych wydarzeń nie będących zderzeniami, ale takich, które może śledzić stworek WHEN.

Numer kodowy	Nr żółwia	Nr pióra	Opis
0	0	0	
1	0	1	
2	0	2	
3			naciśnięcie joystick'a
4	1	0	
5	1	1	
6	1	2	
7			raz na sekundę
8	2	0	
9	2	1	
10	2	2	
11			nie stosowany
12	3	0	
13	3	1	
14	3	2	
15			pozycja joystick'a uległa zmianie
16	3	0	
17	3	1	
18	3	2	
19	0	1	
20	0	2	
21	1	2	

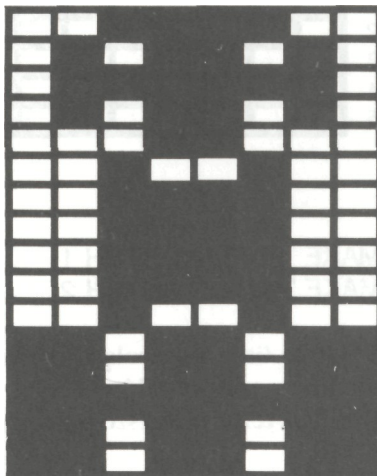
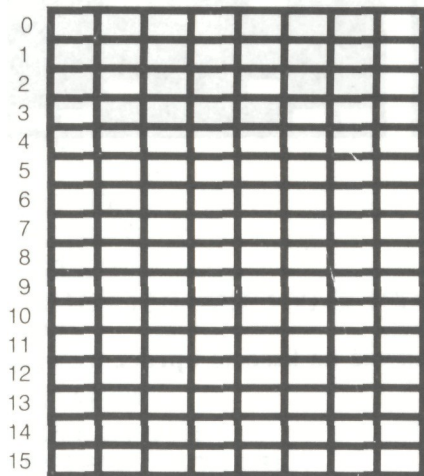
DODATEK C

Zawiera on animowany program graficzny pokazujący sposób wykorzystania niektórych naprawdę wspaniałych możliwości ATARI LOGO. Tworzy on wyścig dwóch samochodów, których prędkości są kontrolowane przez Joystick (jeżeli go nie masz, możesz spróbować zaadaptować sterowanie z grą opisaną w Rozdziale 17).

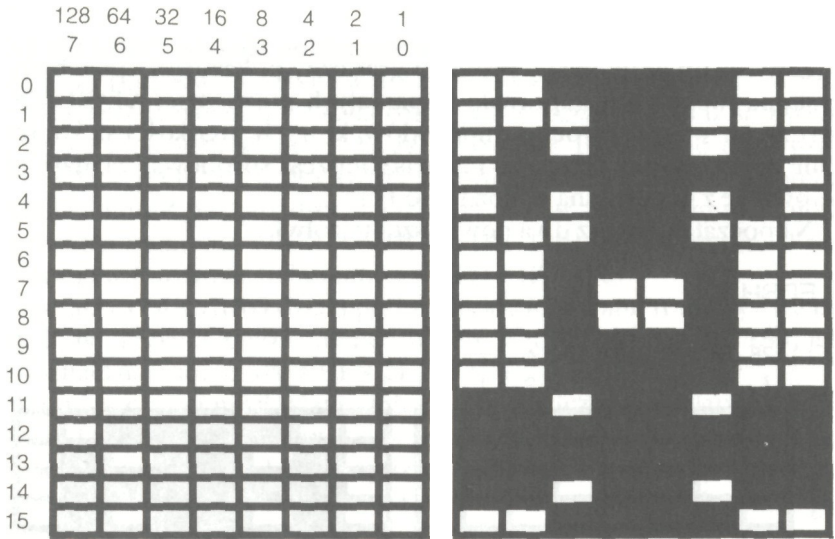
Na początek stwórz dwa nowe kształty żółwi,

EDSH 1

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0



EDSH 2



```
MAKE "AUTO1 GETSH 1  
MAKE "AUTO2 GETSH 2
```

Te dwie linie pozwolą Ci przechować na kasecie oprócz treści gry, także obrazy samochodów.

Przepisz teraz poniższe procedury:

```
TO WYSCIG  
TRASA  
ZDERZENIE  
STEROWANE  
END
```

```
TO TRASA  
TELL 0 HT  
FS  
GRACZ1  
GRACZ2
```


TOR
META
AUTA
END

TOGRACZ1
TELL 1
PUTSH 1 :AUTO1
PU ST
SETSH 1
SETPOS [-30 -100]
END

TO GRACZ2
TELL 2
PUTSH 2 :AUTO2
PU ST
SETSH 2
SETPOS [30 -100]
END

TO TOR
SETBG 0
TELL 0
SETSP 150 WAIT 100 SETSP 0
END

TO META
TELL 3 PU HT
SETPOS [-50 115]
PD
RT 90 SETPN 0
FD 100
SETH 0
END

TO AUTA
TELL 1
SETC 7
TELL 2



SETC 27
END

TO ZDERZENIE
WHEN OVER 1 0 [SETBG BG + 1 IF BG = 1
27 [SETBG 0]
WHEN OVER 2 0 [SETBG BG+ 1 IF BG = 1
27 [SETBG 0]
END

TO STEROWANIE
GAZ 1
HALAS1
GAZ2
HALAS 2
STEROWANIE
END

TO GAZ 1
TELL 1
SETSP (JOY 0) * 5
IF JOYB 0 [SETPOS [-30 -100]]
END

TO HALAS1
TOOT 0 (JOY 0) * 10 + 30 15 4
END

TO GAZ2
TELL 2
SETSP (JOY 1) * 5
IF JOYB1 [SETPOS [30 - 100]]

TO HALAS2
TOOT 1 (JOY 1) * 10 + 30 15 4
END





04-118 WARSZAWA ul. Ostrobramska 128
tel. 100-061 w. 203, tel./fax: 10-28-08

SPECJALISTYCZNY SALON KOMPUTEROWY

oferuje:

Komputery

**ATARI, COMMODORE,
AMIGA, PC XT/AT**

magnetofony, stacje dysków, monitory monochromatyczne
i kolorowe.

Najbogatszą ofertę oprogramowania na kasetach, dyskietkach i
cartridge'ach w formie gotowych zestawów.

Skanery, myszki, pióra świetlne,

Dyskietki 5.25" i 3.5"

Antystatyczne pudełka na dyskietki,

Pokrowce na komputery,

Joysticki.

Emulator hardware'owy komputera:

PC na Atari ST,

PC na Amigę

Największy wybór literatury i czasopism komputerowych.

**Naszą ofertę sprzętową wzbogacamy o poradnictwo i
fachową obsługę popartą 5-letnim doświadczeniem
pracy na komputerach.**

**Zapraszamy do współpracy hurtowników,
sklepy komputerowe, księgarnie.**

OFERTA WYDAWNICZA

Nakładem naszej Agencji ukazały się następujące pozycje na temat ATARI:

1. Atari 65/130 XE - podręcznik użytkownika
2. Atari Basic dla dzieci
3. Grafika Atari
4. Logo Atari

Nakładem naszej Agencji ukazały się następujące pozycje na temat Commodore:

1. Basic dla dzieci i rodziców
2. Podręcznik użytkownika C-64
3. Amiga 500 - podręcznik użytkownika
4. Nieśmiertelności w grach
5. C-64 - Opisy gier

WYDAWCA:



Agencja Wydawnicza
W-wa, tel. 643 25 09