

```

t = 0
DO WHILE klaw$ = ""
    ampl = 90 * 2 ^ (-0.05 * t)
    PSET (INT(160 + ampl * COS(2 * t)), INT(100 + ampl * SIN(t)))
    REM
    REM tu mogą być instrukcje spowalniające wykonywanie programu
    REM
    PRESET (INT(160 + ampl * COS(2 * t)), INT(100 + ampl *
                                                    SIN(t)))

    t = t + .01
    klaw$ = INKEY$
LOOP
SCREEN 0
PRINT "Czas końcowy = "; t
END

```

Program będzie działał dopóty, dopóki nie naciśniesz jakiegoś klawisza.

Ćwiczenie 9-31

A. Zmodyfikuj program QB-32:

- zmieniając liczbę powtórzeń pętli FOR;
- usuwając instrukcję PRESET.

Wybierz taki wariant programu, który uważasz za odpowiedni.

B. Zmodyfikuj wybraną wersję programu QB-32 mnożąc argumenty funkcji SIN i COS przez różne liczby całkowite (np. argument funkcji sinus przez 2, a cosinus przez 3). Możesz ewentualnie zmienić ponadto SIN na COS lub odwrotnie.

9.12. Odczytywanie i zapisywanie danych do plików dyskowych

9.12.1. Umożliwianie dostępu do plików

Plik przeznaczony do wykonania na nim operacji należy otworzyć instrukcją OPEN, wskazując nazwę pliku, rodzaj operacji i umowny numer pliku (od 1 do 255). Do podstawowych operacji należy:

- odczytywanie pliku od początku (OPEN FOR INPUT — otwórz dla wprowadzania),
- zapisywanie (istniejącego lub nowo utworzonego) pliku od początku (OPEN FOR OUTPUT — otwórz dla wyprowadzania)
- dopisywanie danych do końca istniejącego pliku (OPEN FOR APPEND — otwórz dla dołączania).

Numer pliku wskazuje się poprzedzając go słowem AS (jako) i symbolem #. Tak więc pełna przykładowa instrukcja otwarcia pliku DANE1.DAT do zapisu ma postać

```
OPEN "dane1.dat" FOR OUTPUT AS # 2
```

Nazwa pliku w instrukcji OPEN może być określona zmienną łańcuchową, może być zatem wprowadzona przez użytkownika programu.

Otworzony plik zamyka się instrukcją CLOSE z numerem pliku poprzedzonym symbolem #, np. CLOSE #2. Wprawdzie system zamyka pliki kończąc wykonywanie programu, niemniej zaleca się, żeby plik zamykać instrukcją CLOSE, kiedy tylko zapisywanie danych zostało zakończone (bez czekania na koniec programu), zmniejsza to bowiem niebezpieczeństwo przypadkowej utraty danych.

Uwaga. Numery służą do rozróżniania plików, które mogłyby być otwarte jednocześnie. Jeżeli plik, któremu nadaje się numer 2, zostaje zamknięty, a w dalszej części programu jest otwierany jakiś inny plik, to wolno mu przydzielić ten sam numer 2 (już nie zajęty). Jednak używanie w programie jednego numeru na oznaczenie dwóch różnych plików zwiększa szansę popełnienia błędu. Przyjmij zatem zasadę stosowania różnych numerów dla różnych plików.

9.12.2. Wyprowadzanie wyników do pliku

System QBasic zawiera wiele różnych instrukcji służących do zapisywania danych do różnego typu plików. Ograniczymy się tu do jednego typu pliku, tzw. pliku o dostępie sekwencyjnym, w którym dane odczytuje się i zapisuje po kolei, i do jednej instrukcji: PRINT. W razie potrzeby wiadomości te możesz uzupełnić samemu.

Instrukcja PRINT umożliwia zapisywanie wyników bezpośrednio do pliku dyskowego. Od instrukcji pisanie na ekranie różni się tylko tym, że pomiędzy słowem PRINT a listą drukowanych elementów jest podany numer (otworzonego do zapisu) pliku z symbolem # zakończony przecinkiem.

Możesz zmodyfikować dowolny ze swoich programów w taki sposób, żeby wyniki zapisywał w pliku. Przed pierwszą z instrukcji PRINT musisz umieścić instrukcję otwarcia pliku do zapisu, a za ostatnią — instrukcję zamknięcia pliku.

Ćwiczenie 9-32

- A. Zmodyfikuj program QB-5 w taki sposób, żeby zapisywał ponadto w pliku QB05.WYN komunikat pokazywany na ekranie. Czy Twój program jest podobny do programu QB-33?

Program QB-33

```
REM Program QB-33 - zapis do pliku (zmodyfikowany program QB-5)
CONST PI = 3.14159
CLS
INPUT "Podaj promień kola ", r
OPEN "qb05.wyn" FOR OUTPUT AS #2
PRINT "Obwód kola o promieniu"; r; "wynosi"; 2 * PI * r
PRINT #2, "Obwód kola o promieniu"; r; "wynosi"; 2 * PI * r
CLOSE #2
```

- B. Upewnij się, że w katalogu roboczym nie istnieje już plik o nazwie QB05.WYN (jeżeli jest, to usuń go). Uruchom program i sprawdź, że w katalogu powstał plik o tej nazwie zawierający komunikat programu z wprowadzonym przez Ciebie promieniem koła i obliczonym przez program wynikiem. Aby sprawdzić, czy istnieje już plik QB05.WYN i obejrzeć jego zawartość nie musisz wychodzić z systemu QBasic, lecz możesz otworzyć plik za pomocą edytora systemu (File, Open) podając oczywiście nazwę pliku.
- C. Wykonaj ponownie polecenia z punktu B, zmieniając wprowadzaną liczbę. Sprawdź, czy plik zawiera oba komunikaty programu czy jeden, a jeżeli jeden, to czy ostatni czy poprzedni.
- D. Zmodyfikuj program QB-33 (na QB-33a) zmieniając instrukcję otwierania pliku z OPEN FOR OUTPUT na OPEN FOR APPEND. Uruchom program kilkakrotnie i sprawdź zawartość pliku tak jak w punkcie C.
- E. Zastanów się nad zmodyfikowaniem w podobny sposób programu QB-17 (wyznaczającego NWD i NWW dla danych dwóch liczb) w celu zgromadzenia w pliku pewnej ilości wyników. Zwróć uwagę, że (w odróżnieniu od programu QB-5) wprowadzane dane nie są drukowane instrukcją PRINT. Jeżeli uważasz to za przydatne, to dodaj instrukcję drukowania do pliku danych wprowadzonych przez użytkownika.

9.12.3. Odczytywanie danych z pliku

System QBasic zawiera wiele różnych instrukcji i funkcji służących do odczytywania danych z różnego typu plików. Ograniczymy się tu do jednego typu pliku — sekwencyjnego — oraz do jednej instrukcji: INPUT i jednej funkcji: INPUT\$.

Za pomocą funkcji INPUT\$ odczytujesz kolejne znaki z pliku, pojedynczo lub większymi porcjami, natomiast za pomocą instrukcji INPUT odczytujesz od razu całe liczby i łańcuchy.

Ponieważ INPUT\$ jest funkcją, dlatego do nadawania zmiennym wartości należy posłużyć się instrukcją przypisania; w instrukcji INPUT zmienne wymienia się po jej nazwie, tak jak przy odczytywaniu z klawiatury.

Odczytywanie pliku znak po znaku

Do odczytywania znaków należy posłużyć się funkcją INPUT\$. Ma ona dwa argumenty: liczbę znaków odczytywanych jednocześnie (liczba całkowita) oraz numer pliku odczytywanego (z symbolem #), a jej wartością jest łańcuch odczytanych znaków.

Ćwiczenie 9-33

- A. Do odczytywania pliku znak po znaku posłuż się programem QB-34. Odczytaj go i przeanalizuj jego działanie.

Program QB-34

```
REM Program QB-34 - odczytywanie pliku znak po znaku
INPUT "Podaj nazwe pliku ", NazwaPliku$
OPEN NazwaPliku$ FOR INPUT AS #3
NumerZnaku& = 0
CLS
DO WHILE (NOT EOF(3))
    NumerZnaku& = NumerZnaku& + 1
    x$ = INPUT$(1, #3)
    PRINT NumerZnaku&, ASC(x$), x$
LOOP
CLOSE #3
PRINT
PRINT "Odczytano"; NumerZnaku&; "znakow"
END
```

Instrukcję odczytywania kolejnych znaków objęto pętlą, która jest przerywana, kiedy nastąpi koniec pliku (funkcja EOF(3) przybiera wtedy wartość logiczną Prawda, a więc NOT EOF(3) — wartość logiczną Fałsz).

- B. Uruchom program i odczytaj za jego pomocą plik QB05.WYN. Czy zdążysz odczytać na ekranie wszystkie znaki?

Zastanów się, co byś zmienił w programie QB-34, żeby umożliwiał odczytanie na ekranie wszystkich znaków.

Wykonaj teraz następne ćwiczenie i porównaj swoje uwagi i propozycje z modyfikacjami programu wprowadzanymi w tym ćwiczeniu.

Ćwiczenie 9-34

- A. Zmodyfikuj program QB-34 w taki sposób, żeby po odczytaniu takiej ilości znaków, która powinna zappełnić ekran (np. 20), zatrzymywał się i czekał na naciśnięcie:
- wyróżnionego klawisza w celu zakończenia programu,
 - dowolnego innego klawisza w celu kontynuowania odczytu pliku.

Uwaga. Do sprawdzenia, czy liczba znaków osiągnęła kolejną wielokrotność zadanej liczby, możesz posłużyć się dzieleniem całkowitym (rozdział 9.5.1).

- Porównaj swój program z programem zapisanym w pliku QB34M.BAS.
- B. Rozpoznaj w odczytywanym pliku znaki końca linii.
- C. Za pomocą programu QB-34 lub programu zmodyfikowanego przez Ciebie odczytaj plik ZNST1 i porównaj wyniki z przedstawionymi na rys. 4.23.
- D. Odczytaj (krótki) plik z polskimi znakami (np. VGA.TAG lub jeden z pozostałych plików znajdujących się na załączonej dyskietce). Zwróć uwagę na kody znaków. Czy możesz rozpoznać (sprawdzić) standard kodowania?

Odczytywanie łańcucha znaków

Odczytywałeś już zmienne łańcuchowe posługując się instrukcją INPUT. System traktował naciśnięcie przez Ciebie klawisza *Enter* jako zakończenie łańcucha. Zauważ, że to samo dotyczyło wprowadzania kilku liczb. Podobnie jest przy odczytywaniu danych z pliku. System traktuje napotkane w nim znaki końca linii (zazwyczaj jest to para znaków o kodach 13 i 10) jako koniec łańcucha. Koniec pliku możesz rozpoznawać podobnie jak w programie QB-34.

Ćwiczenie 9-35

- A. Napisz program odczytujący z pliku kolejne łańcuchy znaków. Czy Twój program jest podobny do programu QB-35?

Program QB-35

```
REM Program QB-35 - odczytywanie zmiennych łańcuchowych z pliku
CLS
INPUT "Podaj nazwę pliku ", NazwaPliku$
OPEN NazwaPliku$ FOR INPUT AS #3
NumerLancucha& = 0
DO WHILE (NOT EOF(3))
    NumerLancucha& = NumerLancucha& + 1
```

```
INPUT #3, lancuch$
PRINT NumerLancucha&, lancuch$
LOOP
CLOSE #3
PRINT
PRINT "Odczytano"; NumerLancucha&; "lancuchow"
END
```

- B. Uruchom program i odczytaj zawartość pliku QB05.WYN.
- C. Odczytaj zawartość pliku PLIKDIR.TXT utworzonego poleceniem systemu operacyjnego `DIR > plikdir.txt`. Jeżeli drukowany na ekranie numer łańcucha przeszkadza Ci, to usuń go z instrukcji.

Odczytywanie liczb i tekstów

Z pliku zawierającego dane można odczytywać informacje większymi porcjami, a nie tylko znak po znaku. W szczególności można odczytywać łańcuchy znaków oraz liczby. Pewnym problemem jest zakończenie odczytywania danych z pliku we właściwym momencie; jeżeli się tego nie uczyni, to program się zatrzyma sygnalizując błąd. Wygodnie jest znać liczbę odczytywanych elementów. Jeżeli liczba elementów nie jest znana, to można postąpić w sposób analogiczny do stosowanego przy odczytywaniu pojedynczych znaków (sprawdzanie wartości funkcji EOF).

Można w ten sposób odczytywać:

- teksty do zmiennych łańcuchowych i liczby do zmiennych numerycznych;
- kilka liczb umieszczonych w jednym wierszu (trzeba w instrukcji `INPUT` uwzględnić w tym celu odpowiednią liczbę zmiennych);
- dane przygotowane w taki sposób, że po jednej lub kilku liczbach w tym samym wierszu jest umieszczony tekst (w takiej sytuacji ostatnia ze zmiennych na liście instrukcji `INPUT` musi być zmienną tekstową).

Podczas odczytywania pliku zawierającego wiele wierszy danych, dane te można umieszczać w odpowiednich tablicach.

Plik o nazwie `DANE2.DAT`, znajdujący się na załączonej dyskietce w katalogu `PLIKI\DANE_DAT`, zawiera kilka wierszy danych, przy czym w każdym wierszu jest kolejno: liczba rzeczywista, liczba całkowita oraz tekst (imię), np.

```
34.56 132 Krzysztof
```

Dane te można odczytać za pomocą programu zawierającego instrukcję podobną do:

```
INPUT #3, LiczbaRzeczywista, LiczbaCalkowita%, Lancuch$
```


Ćwiczenie 9-36

- A. Napisz własny program odczytywania danych o przedstawionej postaci do tablic i uruchom go (lub odczytaj program QB-35), a następnie odczytaj plik DANE2.DAT z załączonej dyskietki. Porównaj otrzymane wyniki z zawartością pliku.
- B. Zmodyfikuj program QB-35 w taki sposób, żeby mógł odczytywać pliki zawierające do 60 liczb całkowitych zapisanych po jednej w wierszu. Sprawdź jego działanie odczytując plik DANE3.DAT.
- C. Zmodyfikuj program QB-35 w taki sposób, żeby mógł odczytywać pliki zawierające po 4 liczby w wierszu, przy czym druga z liczb jest rzeczywista, a pozostałe są całkowite. Sprawdź jego działanie odczytując plik DANE4.DAT.
- D. Zmodyfikuj program QB-35 w taki sposób, żeby mógł odczytywać pliki zawierające w każdym wierszu liczbę całkowitą oraz tekst. Odczytywane liczby umieszczaj w tablicy *odpowiedz*%, a łańcuchy w tablicy *pytanie*\$. Sprawdź działanie programu odczytując plik DANE5.DAT.

9.13. Przykładowe programy

Możesz już napisać wszystkie programy odpowiadające zadaniom z rozdziału 8. Omówimy tu przede wszystkim koncepcję ich utworzenia, dodając niekiedy (fragmentaryczne) wyjaśnienia. Jeżeli w rozdziale 8 został przedstawiony algorytm rozwiązania zadania, to potraktuj tworzenie programu jako dodatkowe ćwiczenie z tzw. kodowania programu za pomocą instrukcji języka programowania.

Program przekształcania pliku — zadanie 8-13

Zadanie 8-13 dotyczy odczytania parzystej ilości liczb zapisanych w pliku po jednej w linii i zapisania ich w innym pliku po dwie obok siebie. Znasz już wszystkie potrzebne instrukcje na tyle dokładnie, że powinieneś poradzić sobie z napisaniem programu bez dodatkowych wskazówek. Napisz go i sprawdź jego działanie na danych zapisanych w pliku DANE6.DAT. Możesz porównać swój program z programem zapisanym w pliku QBDWLICZ.BAS.

Program testu — według algorytmu 15 do zadania 8-11

Schemat postępowania z wybieraniem pytań testowych z tablicy jest przedstawiony w algorytmie 15, napisanie zasadniczego zrzębu programu nie powinno przedstawiać zatem trudności. Pewną trudność może sprawić to,

że w systemie QBasic nie ma zmiennych logicznych. Wartości logiczne są w nim reprezentowane za pomocą liczb całkowitych: do reprezentowania wartości logicznej Fałsz służy 0, a do wartości Prawda dowolna liczba różna od zera (zazwyczaj -1).

Odczytywanie danych możesz oprzeć na programie opracowanym w punkcie D ostatniego ćwiczenia (możesz przekształcić go na procedurę, lecz nie musisz). Do testowania programu możesz użyć pliku DANETEST.DAT. Porównaj teraz swój program z programem QB-TEST; (zawartym w pliku QBTEST.BAS) lub z jego zasadniczym fragmentem przedstawionym poniżej jako QB-36.

Program QB-36

```
DIM zadane%(1 TO 40), pytanie$(1 TO 40), odpowiedz%(1 TO 40)
RANDOMIZE TIMER
REM instrukcje odczytania danych (lub wywołanie procedury)
REM krok 1
dobreodp% = 0
numer% = 1
numerpom% = 1
REM krok 2
FOR numerpom% = 1 TO 40
    zadane%(numerpom%) = 0
NEXT numerpom%
REM krok 3
DO
    DO
        numerlos% = 1 + INT(40 * RND)
    LOOP UNTIL zadane%(numerlos%) = 0
REM krok 4
PRINT pytanie$(numerlos%)
zadane%(numerlos%) = -1
REM Krok 5
INPUT "Podaj wybrany numer odpowiedzi ", numerodp%
IF numerodp% = odpowiedz%(numerlos%) THEN dobreodp% =
    dobreodp% + 1
REM Kroki 6 i 7
IF numer% < 20 THEN
    INPUT "Czy chcesz odpowiedzieć na następne pytanie?
        Tak/Nie ", nast$
    nast$ = LCASE$(nast$)
    IF nast$ = "tak" OR nast$ = "t" THEN nast$ = "tak"
    IF nast$ = "tak" THEN numer% = numer% + 1
ELSE
    nast$ = "nie"
```



```
END IF
LOOP WHILE nast$ = "tak" AND numer% <= 20
REM Krok 8
PRINT "Udzieliles"; dobreodp%; "prawidlowych odpowiedzi na";
PRINT numer%; "pytan"
END
```

Program QB-TEST jest napisany w taki sposób, że odpowiedniość między fragmentami programu a krokami algorytmu jest dobrze widoczna. Program możesz samodzielnie ulepszyć, np. zmienić sposób odczytywania odpowiedzi użytkownika (INKEY\$ zamiast INPUT) oraz zmienić sposób losowania zgodnie z uwagami podanymi w rozdz. 8. Podawanie nazwy pliku możesz zastąpić wybieraniem nazwy przedmiotu. Liczba pytań do różnych przedmiotów nie musi być identyczna. Możesz także umożliwić ładniejsze przedstawianie na ekranie odpowiedzi do wyboru — w oddzielnych liniach — przez osobne umieszczenie tych odpowiedzi w pliku danych (każda w oddzielnej linii) i oddzielne umieszczanie ich w tablicy (mogłaby zostać użyta jedna tablica z dodatkowym indeksem, oznaczającym numer odpowiedzi). Liczba odpowiedzi do wyboru nie musiałaby być identyczna dla wszystkich pytań; musiałbyś jednak podawać w pliku danych, i zapamiętywać w programie liczbę odpowiedzi do poszczególnych pytań. Możesz także utworzyć procedury realizujące poszczególne czynności.

Program sporządzania mapy funkcji — zadanie 8-10

Koncepcja odwzorowywania funkcji dwóch zmiennych na ekranie z użyciem kolorów została przedstawiona w rozdz. 8. Trzeba wybrać taki tryb graficzny, który umożliwia stosowanie kilku kolorów, np. SCREEN 7 (opis trybów graficznych znajduje się w pomocy pod hasłem SCREENmodes). Z rozdzielczości ekranu w wybranym trybie wynika sposób wzajemnego powiązania zmiennych niezależnych x i y funkcji ze współrzędnymi ekranu, np. przy rozdzielczości równej 320 pikseli zmienna x z zakresu $0 \leq x < 10$ powinna być mnożona przez 32 (i zmniejszana do wartości całkowitej). Do przyporządkowania kolorów trzeba użyć instrukcji warunkowych. Możesz porównać sporządzony przez siebie program z programem zapisanym w pliku QBMAPA.BAS.

Czas działania programu przy obliczaniu wartości koloru oddzielnie dla każdego punktu ekranu może być znaczny. Jeżeli tak się okaże, to zmniejsz odpowiednio liczbę punktów w poziomie i pionie, pokrywając obrazem jedynie część ekranu. Spróbuj teraz zmienić granice przedziałów odpowiadających poszczególnym kolorom, a także, jeśli pozwala na to karta graficzna, użyć większej liczby kolorów.

Programy zmiany kodów znaków w pliku — zadanie 8-12

Pisząc te programy możesz skorzystać z programu QB-34 do odczytywania pliku znak po znaku. Do szyfrowania możesz użyć prostego algorytmu zmniejszającego wartość kodu o 1 lub o wielkość podaną przez użytkownika. (Tekst „To jest szyfr” po zmniejszeniu kodów o 1 wyglądałby następująco: „Sn idrs ryxex”. A jak byś odszyfrował słowo: „Jnlotsdq”?) Uruchamiając program sprawdź m.in., czy zaszyfrowany plik po odszyfrowaniu jest identyczny z plikiem oryginalnym. Do zamiany standardu polskich znaków użyj instrukcji SELECT CASE (rozdz. 9.6.3). Możesz porównać sporządzone przez siebie programy z programami zapisanymi w plikach QBSZYFR.BAS i QBMAZISO.BAS (zamiana standardu z kodu Mazovia na ISO Latin 2). Kilka plików w kodzie Mazovia znajduje się na dyskietce.

Program do porządkowania liczb — zadanie 8-4

Znasz już elementy umożliwiające Ci samodzielne napisanie programu. Liczby umieścisz w tablicy. Program wyznaczający największą liczbę oparty na algorytmie 5 możesz przerobić na procedurę lub umieścić w pętli. Podobnie algorytmy przesuwania liczb w nieuporządkowanej części tablicy. Program swój możesz porównać z programem zapisanym w pliku QBSZEREG.BAS. Wskazówka: na etapie uruchamiania programu posłuż się instrukcjami drukowania pomocniczych zmiennych, np. indeksów tablicy.

Program do dodawania „pisemnego” — zadanie 8-8

Dodawanie „pisemne”, tak proste na papierze, może Ci sprawić nieco trudności. Musisz zdecydować się, jak chcesz reprezentować poszczególne cyfry w pamięci programu. Zauważ, że nie są to wyłącznie cyfry, lecz także „miejsca puste”, traktowane przy dodawaniu jako zera. Naturalną strukturą odpowiadającą papierowi kratkowanemu jest tablica dwuwymiarowa. Jej elementami mogą być liczby całkowite (do przedstawiania cyfr posłużą liczby od 0 do 9, a do zapamiętania miejsca pustego np. liczba ujemna). Można też posłużyć się zmiennymi łańcuchowymi. Zapisanie w tablicy wprowadzonych przez użytkownika liczb, które należy dodać, wymaga rozbięcia liczb wielocyfrowych na liczby jednocyfrowe — oznaczające pozycje jedności, dziesiątek, setek itd. Do tego celu należy posłużyć się dzieleniem całkowitym. Pisząc swój program, możesz przyjrzeć się rozwiązaniom zastosowanym w programie zapisanym w pliku QBDODPIS.BAS.

Jeśli chcesz, to po uruchomieniu swojego programu lub przeanalizowaniu działania programu z dyskietki zastanów się, co należałoby w nim zmienić, żeby realizował (ilustrował) odejmowanie liczb. A może jest potrzebny program ilustrujący wykonywanie innych działań arytmetycznych ...?