**Множества**

Понятие множества в Паскале очень близко к математическому определению: множество - это совокупность однотипных неиндексированных объектов. Множества описываются в виде:

SET OF тип ,

где тип - базовый для этого множества тип, т.е. тип элементов множества. Базовый тип должен быть порядковым типом мощностью не более 256 (т.е. допускающий не более 256 различных значений), причем порядковые номера (вспомним функцию ORD) наименьшего и наибольшего значений должны лежать на отрезке [0,255]. Таким образом, базовым типом для множества могут быть: типы Char, Boolean, Byte и все производные от Byte интервальные типы. Размер объекта типа “множество” можно определить по формуле: размер = (мощность-1) DIV 8 + 1, т.е. множества - довольно компактные объекты, самое большое множество имеет размер 32 байта. Неименованные константы типа множество записываются в виде:

[ подмножество , подмножество , ... ] ,

где подмножество - это либо отдельное значение, либо диапазон. Диапазон записывается как начальное значение .. конечное значение. Любое из значений может быть как константой, так и выражением соответствующего типа. Запишем, например, константу-множество, содержащую числа 0, 1, 2, 3, 4, 8, 12, 13, 14, 15, 16, 22:

[0,1,2,3,4,6,12,13,14,15,16,22]

или

[0..4,6,12..16,22]

или

[0..2,3..4,6..6,12,13..16,22]

или

[22,13..15,1..6,0,12,16]

Все эти константы полностью эквивалентны, порядок записи элементов совершенно произволен. Допускаются пустые множества, они записываются так: [ ]. Опишем несколько переменных и типизированных констант:

TYPE MySet = SET OF 0..111;

VAR a : SET OF Char;

b : MySet;

CONST c : MySet = [];

d : SET OF Char = ['А'..'Я'];

e : SET OF Boolean = [FALSE];

К множествам применимы следующие операции:

- множеству можно присвоить множество того же типа;

- операция объединение +

- операция дополнение -

- операция пересечение \*

- операция эквивалентность =

- операция не эквивалентность <>

- операция включение <= и >=

Последние три операции дают значения логического типа - TRUE или FALSE. Пусть множество A=[1..20] , B=[2..9,15..20] , C=[3..22] , тогда A+B=[1..20] , A+C=[1..22], A-C=[1,2], C-A=[21,22], A\*B=[1..20], A\*C=[3..20], B\*C=[3..9,15..20] , A=B =FALSE , A<>C =FALSE , B<=A =TRUE , A>=C =FALSE.

Существует еще одна операция, связанная с множествами, - операция IN, она применяется к скалярной величине и множеству:

выражение IN множество

Здесь выражение - любое выражение базового для данного множества типа, результат операции - TRUE, если такой элемент есть в множестве, и FALSE - в противном случае.

Для множеств определены две стандартные процедуры:

Include ( множество , выражение )

Exclude ( множество , выражение )

Процедура Include добавляет элемент, равный значению выражения в множество, а процедура Exclude удаляет такой элемент из множества.

Теперь, когда мы знаем все возможности множеств, запишем программу, находящую все простые числа на отрезке [1,255] :

{программа использует алгоритм "решето Эратосфена"}

TYPE NumSet = SET OF 1..255;

CONST N : NumSet=[2..255]; { исключили 1 как не являющуюся простым числом }

VAR MaxDivider,d : Byte; k : Word;

BEGIN MaxDivider:=Round(SQRT(255));

d:=2;

WHILE d<=MaxDivider DO BEGIN

k:=2\*d;

WHILE k<=255 DO BEGIN Exclude(N,k); INC(k,d); END;

INC(d);

END;

WRITELN('Простые числа :');

FOR k:=1 TO 255 DO IF k IN N THEN WRITE(k:4);

END.

Решим еще одну задачу : ввести массив символов и подсчитать, сколько в нем русских и латинских букв.

TYPE Letters = SET OF Char;

CONST Lat = ['a'..'z','A'..'Z']; Rus = ['а'..'п','р'..'я','А'..'Я'];

VAR c : Char;

CONST RSum : Word=0; LSum : Word=0;

BEGIN WRITE('Введите массив символов, затем нажмите Enter ');

REPEAT READ(c);

IF c IN Lat THEN INC(LSum) ELSE IF c IN Rus THEN INC(RSum);

UNTIL c=#10;

WRITELN('Латинских букв ',LSum,' русских букв ',RSum);

END.

Обратите внимание, что в этой задаче нет необходимости заранее знать, сколько символов содержится в массиве (более того, в программе никакого массива и нет!), достаточно лишь помнить, что клавиша Enter генерирует символ #10.