# **Лабораторная работа** 20

#### 20***.1 [#***3***0]***

Разработайте с использованием типа данных множество программу

PROGRAM Prime(INPUT, OUTPUT)

которая находит простые числа в  диапазоне, ограниченном от 2 до 100, методом «Решето Эратосфена», который заключается в следующем:

Принимается, что минимальное простое число равно 2. В «решето» помещают все числа в заданном диапазоне. Минимальное число в диапазоне считается простым. Далее оно удаляется из решета и удаляются все ему кратные. Новое минимальное число в диапазоне также является простым.   
Процесс повторяется, пока не будет достигнуто максимальное число в диапазоне.

Для диапазона 2 .. 16 процесс будет выглядеть следующим образом.  
S = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}  
удаляем кратные 2  
S – {2, 4, 6, 8, 10, 12, 14, 16} = {3, 5, 7, 9, 11, 13, 15}  
удаляем кратные 3  
{3, 5, 7, 9, 11, 13, 15} – {3, 6, 9, 12, 15} = {5, 7, 11, 13}  
и так далее…

Пример

Содержимое решета удаляем кратное

{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16} {2,4,6,8,10,12,14,16} 2

{3,5,7,9,11,13,15} {3,6,9,12,15} 3

{5,7,11,13} {5,10,15} 5

{7,11,13} {7,14} 7

{11,13} {11} 11

{13} {13} 13

Простые числа в диапазоне до 16 будут 2,3,5,7,11,13

#### 20***.2 [#***3***0]***

Соберите программу SortDate. Допишите недостающие разделы проекта. Оформите процедуры работы с датами в виде модуля.

Входной файл дат называется 'FI.TXT'.

DP1

PROGRAM SortDate(INPUT, OUTPUT);

TYPE

Month = (NoMonth, Jan, Feb, Mar, Apr, May, Jun,

Jul, Aug, Sep, Oct, Nov, Dec);

DayNum = 1..31;

Date = RECORD

Mo : Month;

Day: DayNum

END;

FileOfDate = FILE OF Date;

VAR

Copying: BOOLEAN;

D, VarDate: Date;

TFile, DateFile: FileOfDate;

FInput: TEXT;

BEGIN{SortDate}

ASSIGN(DateFile, 'DF.DAT');

ASSIGN(TFile, 'TF.DAT');

ASSIGN(FInput, 'FI.TXT');

REWRITE(DateFile);

RESET(FInput);

ReadDate(FInput, VarDate);

READLN(FInput);

WRITE(DateFile, VarDate);

RESET(DateFile);

WHILE NOT EOF(FInput)

DO

{Поместить новую дату в DateFile в соответствующее место}

{Копируем DateFile в OUTPUT}

END.{SortDate}

PROCEDURE WriteDate(VAR FOut: TEXT; VAR Res: Date);

BEGIN{WriteDate}

WriteMonth(FOut,Res.Mo);

WRITE(FOut,Res.Day:3)

END;{WriteDate}

PROCEDURE ReadDate (VAR FIn: TEXT; VAR Res: Date);

BEGIN{ReadDate}

ReadMonth(FIn,Res.Mo);

READ(FIn,Res.Day)

END;{ReadDate}

PROCEDURE ReadMonth(VAR FIn: TEXT; VAR Mo: Month);

VAR

Ch1, Ch2, Ch3: CHAR;

BEGIN{ReadMonth}

READ(FIn, Ch1, Ch2, Ch3);

IF (Ch1 = 'J') AND (Ch2 = 'A') AND (Ch3 = 'N') THEN Mo := Jan ELSE

IF (Ch1 = 'F') AND (Ch2 = 'E') AND (Ch3 = 'B') THEN Mo := Feb ELSE

IF (Ch1 = 'M') AND (Ch2 = 'A') AND (Ch3 = 'R') THEN Mo := Mar ELSE

IF (Ch1 = 'A') AND (Ch2 = 'P') AND (Ch3 = 'R') THEN Mo := Apr ELSE

IF (Ch1 = 'M') AND (Ch2 = 'A') AND (Ch3 = 'Y') THEN Mo := May ELSE

IF (Ch1 = 'J') AND (Ch2 = 'U') AND (Ch3 = 'N') THEN Mo := Jun ELSE

IF (Ch1 = 'J') AND (Ch2 = 'U') AND (Ch3 = 'L') THEN Mo := Jul ELSE

IF (Ch1 = 'A') AND (Ch2 = 'U') AND (Ch3 = 'G') THEN Mo := Aug ELSE

IF (Ch1 = 'S') AND (Ch2 = 'E') AND (Ch3 = 'P') THEN Mo := Sep ELSE

IF (Ch1 = 'O') AND (Ch2 = 'C') AND (Ch3 = 'T') THEN Mo := Oct ELSE

IF (Ch1 = 'N') AND (Ch2 = 'O') AND (Ch3 = 'V') THEN Mo := Nov ELSE

IF (Ch1 = 'D') AND (Ch2 = 'E') AND (Ch3 = 'C') THEN Mo := Dec

ELSE Mo := NoMonth

END;{ReadMonth}

PROCEDURE WriteMonth(VAR FOut: TEXT; VAR Mo: Month);

VAR

Ch1, Ch2, Ch3: CHAR;

BEGIN {WriteMonth}

IF Mo = Jan THEN WRITE('JAN') ELSE

IF Mo = Feb THEN WRITE('FEB') ELSE

IF Mo = Mar THEN WRITE('MAR') ELSE

IF Mo = Apr THEN WRITE('APR') ELSE

IF Mo = May THEN WRITE('MAY') ELSE

IF Mo = Jun THEN WRITE('JUN') ELSE

IF Mo = Jul THEN WRITE('JUL') ELSE

IF Mo = Aug THEN WRITE('AUG') ELSE

IF Mo = Sep THEN WRITE('SEP') ELSE

IF Mo = Oct THEN WRITE('OCT') ELSE

IF Mo = Nov THEN WRITE('NOV') ELSE

IF Mo = Dec THEN WRITE('DEC')

ELSE WRITE('NoMonth')

END; {WriteMonth}

FUNCTION Less(VAR D1, D2: Date): BOOLEAN;

{Less:= D1 < D2}

BEGIN {Less}

IF D1.Mo < D2.Mo

THEN

Less := TRUE

ELSE

IF D1.Mo > D2.Mo

THEN

Less := FALSE

ELSE {D1.Mo = D2.Mo}

Less := (D1.Day < D2.Day)

END; {Less}

PROCEDURE CopyOut(VAR DateFile: FileOfDate);

VAR

VarDate: Date;

BEGIN {CopyOut}

WHILE NOT EOF(DateFile)

DO

BEGIN

READ(DateFile, VarDate);

WriteDate(OUTPUT, VarDate);

WRITELN;

END

END;{CopyOut}

DP 1.2

{Копируем DateFile в OUTPUT}

RESET(DateFile);

CopyOut(DateFile);

DP 1.1

{Поместить новую дату в DateFile в соответствующее место}

BEGIN

ReadDate(FInput,D);

READLN(FInput);

IF (D.Mo <> NoMonth)

THEN

BEGIN

{копируем элементы меньшие,чем D из DateFile в TFile}

{копируем D в TFile}

WRITE(TFile, D);

{копируем остаток DateFile в TFile}

{копируем TFile в DateFile}

END;

END;

DP 1.1.1

{копируем элементы меньшие, чем D из DateFile в TFile}

BEGIN

REWRITE(TFile);

Copying := TRUE;

WHILE NOT EOF(DateFile) AND Copying

DO

BEGIN

READ(DateFile, VarDate);

IF Less(VarDate,D)

THEN

WRITE(TFile, VarDate)

ELSE

Copying := FALSE

END

END;