Simple Selection

A typical selection statement is

```
if BankBalance <0 then
          write('overdrawn')
else
          write('in credit')
```

it selects `write('overdrawn')`
or `write('in credit')`

according to whether the boolean expression
`BankBalance <0`
is true or false.

Another selection statement, which does something or nothing is

```
if mark < 40 then
          write('Fail')
```

`mark < 40`
`BankBalance <0`
are both examples of boolean expressions. Like all expressions, *boolean* expressions represent values. The simplest examples are the constants of the ordinal type *boolean* :

false true

The `odd(x)` function also returns a *boolean* result, true if it is odd,false if x is even.

We can form longer boolean expressions by using the relational operators:

=	equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
<>	not equal to

Assignment of boolean values

var

```
    BankBalance:real; Student, Rich:boolean; Temperature:integer; NotDivBy2:boolean;
...
Student:=true;
Rich:=BankBalance >= 0;
NotDivBy2 := odd(Temperature)
```

The operands of the relational operators (the values that are being compared) must be of the same type. The exception to this rule is that reals can be compared with integers, but BEWARE the value calculated by the computer of something like

    3.0*(10.00/3.0)

may not be 10.0 exactly.

In Pascal we cannot use the relational operators in the mathematical manner 5<x<10. The Pascal equivalent is

    (5 < x) **and** (x<10)

The boolean operators are **and**, **or** and **not**. These have precedence over the relational operators.

- The if statement has the following format:

    if *boolean expression* **then**
        then statement
    else
        else statement

When an **if** statement is entered, its boolean expression is evaluated. If it is true, the *then statement* is carried out, and *else statement* is skipped. If the statement is false, *else statement* is executed instead.

- It is possible to have an **if** statement without an **else**
  if *boolean expression* **then** *then statement*
- There is never a semicolon immediately before the **else**

Example

```
program TempConv;
{$APPTYPE CONSOLE}
var Prop:Char;
var Fahren, Cent : real;
begin
  write('Input Fahrenheit temperature: ');
  readln(Fahren);
```

```
  Cent := (Fahren - 32) * 5 / 9;
  writeln;
  writeln('Centigrade equivalent is ', Cent);
  if Cent > 100 then
    writeln(' which is higher than boiling point')        // no semicolons allowed here!
  else
    writeln('Which is below or at boiling pt');
  readln(Prop);                                           // this line is always executed
end.
```

Exercises

1. Write a program that reads in two numbers and outputs the smallest of the two.
2. Write a program to print out the square root of an integer. If the integer is less than 0, print a message to the screen.
3. Write a program that will add or remove money from an imaginary account. You will need to have a variable that holds an imaginary balance and you will write this amount at the top of the screen. Ensure that you write the final balance at the end. Use something similar to the following as your input screen:

   Imaginary Account

   Current balance of account is:
   Amount of money to use:

   Options:

   1. Add money to account
   2. Remove money from account

   Enter option here:

   New balance of account is:

4. Update your program so that it sends a message to the screen if the option entered is invalid.
5. Update your program so that it will only remove money if there is enough in the account. If there is not enough money, send a message to the screen.

HWK: Complete class exercises and email them to me by 24th January, 2002.

Further selection statements

Recall that the if statement has the following format:

if *boolean expression* **then**
then statement
// **else**
// *else statement*

- Several statements will be treated as one if they are surrounded by **begin ... end**
- In particular, a *then statement* or an *else statement* can be replaced by several statements, if they are surrounded by **begin ... end**
- The *boolean expression* can be made up from several smaller boolean expressions linked by **and or not.** Extra round brackets may be needed.

Example

```
program TempConv;
{$APPTYPE CONSOLE}
// line is required to run Pascal within Delphi
var Prop:Char;

// this program reads a temperature in Fahrenheit,
// calculates Centigrade equivalent , and outputs it

// Suitable messages are output according to whether the temperature is
 // above or below boiling point

var
  Fahren, Cent : real;
begin
  write('Input Fahrenheit temperature: ');
  readln(Fahren);
  Cent := (Fahren - 32) * 5 / 9;
  writeln;
  writeln('Centigrade equivalent is ', Cent);
  if (Cent > 100) and (Cent < 1000) then
    begin
      writeln(' which is higher than the boiling point of water');
      writeln(' 100 degrees in centigrade');
```

```
    writeln(' but less than 1000')
  end                  / / no semicolons allowed here!
 else
  writeln('Which is below or at boiling pt or VERY hot');
 readln(Prop);
end.
```

- If statements can be nested inside each other, for example, if hot and red are boolean variables

```
if hot then
                if red then
                                writeln('I''m hot & red')
                else
                                writeln('I''m just hot');
    writeln('I''d like some water');
```

- But it is hard to remember which if the else belongs to! Hence it is best to say

```
if hot and red then
     writeln('I''m hot & red')
else if hot then
     writeln('I''m just hot');
writeln('I''d like some water');
```

Consider this code:

```
if num>=0 then
     if num = 0 then
          writeln('Number is 0');
else
     writeln('Number is less than 0');
```

At first glance it looks as if whenever num is less than 0 a message will be output stating this. However, Pascal always associates an *else* with the nearest *if*, so the words 'Number is less than 0' will only be output if the number is equal to 0! We can use *begin* and *end* to sort this out.

```
if num>= 0 then
     begin
     if num = 0 then
          writeln('Number is 0');
     end
else
     writeln('Number is less than 0');
```

Or just simplify the code:

```
if num<0 then
    writeln('Number is less than 0');
else if num=0 then
    writeln('Number is 0');
```

Exercises

17. Write a program which asks for a day of the month, gives error if it is outside range 1..31; then asks for or month number, checks it for validity, then writes out a message only if it is your birthday.
18. Modify your program to calculate the roots of a quadratic so it writes out an error message if there are no real roots; otherwise output the roots as before.
19. Write a program which reads in an angle in the range 0<=angle<360 and prints out the quadrant in which it lies. The output should be of the following form:

    The angle lies in the first quadrant

    If the input angle is outside the given range then print an appropriate message.