PASCAL CASE STATEMENT

http://www.tutorialspoint.com/pascal/pascal_case_statement.htm

You have observed that **if-then-else** statements enable us to implement multiple decisions in a program. This can also be achieved using the **case** statement in simpler way.

Syntax

The syntax of the case statement is −

```
case (expression) of
   L1 : S1;
   L2: S2;
   ...
   ...
   Ln: Sn;
end;
```

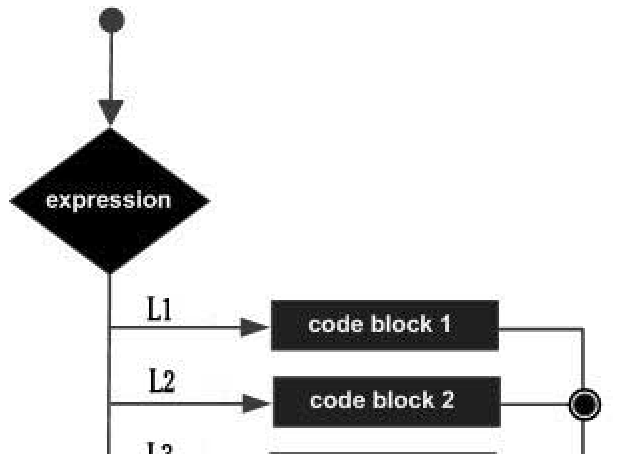
Where, **L1**, **L2**... are case labels or input values, which could be integers, characters, boolean or enumerated data items. **S1**, **S2**, ... are Pascal statements, each of these statements may have one or more than one case label associated with it. The expression is called the **case selector** or the **case index**. The case index may assume values that correspond to the case labels.

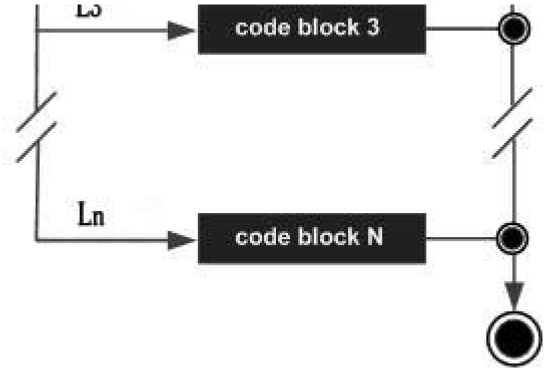
The case statement must always have an **end** statement associated with it.

The following rules apply to a case statement −

- The expression used in a case statement must have an integral or enumerated type or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a case. Each case is followed by the value to be compared to and a colon.
- The case label for a case must be the same data type as the expression in the case statement, and it must be a constant or a literal.
- The compiler will evaluate the case expression. If one of the case label's value matches the value of the expression, the statement that follows this label is executed. After that, the program continues after the final end.
- If none of the case label matches the expression value, the statement list after the else or otherwise keyword is executed. This can be an empty statement list. If no else part is present and no case constant matches the expression value, program flow continues after the final end.
- The case statements can be compound statements *i. e. , aBegin. . . Endblock*.

Flow Diagram





Example

The following example illustrates the concept −

```
program checkCase;
var
   grade: char;
begin
   grade := 'A';

   case (grade) of
      'A' : writeln('Excellent!' );
      'B', 'C': writeln('Well done' );
      'D' : writeln('You passed' );
      'F' : writeln('Better try again' );
   end;

   writeln('Your grade is  ', grade );
end.
```

When the above code is compiled and executed, it produces the following result −

```
Excellent!
Your grade is A
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js