

THE CASE STATEMENT

A simpler way to choose one option when If-Else If would get too long.

1. You already know If

If asks one yes/no question. Else If asks the next question only if the first answer was no.

```
if iChoice = 1 then
    ShowMessage('Add')
else if iChoice = 2 then
    ShowMessage('Edit')
else if iChoice = 3 then
    ShowMessage('Delete')
else
    ShowMessage('Invalid');
```

That works. But look at it: we keep asking the same variable the same kind of question — “Is it this exact value?” Case was made for that pattern.

2. The same choice written as Case

```
case iChoice of
    1: ShowMessage('Add');
    2: ShowMessage('Edit');
    3: ShowMessage('Delete');
else
    ShowMessage('Invalid');
end;
```

Read it as: “Look at iChoice. If it is 1, do Add. If it is 2, do Edit. If it is 3, do Delete. If it is anything else, say Invalid.”

Case does not replace If. It is If’s cousin for one special job: matching one variable against a list of exact options.

3. How the option is selected — very slowly

The learner types a number into a menu, or the program already has a value in a variable. That value is the selector.

- Delphi looks at the selector once.
- It walks down the list: is it 1? is it 2? is it 3?
- The first matching number wins. That branch runs. The others are skipped.
- If nothing matches, the else branch runs — if you wrote one.
- Then Case is finished. The program continues after end;

So the “option” is not chosen by magic. The user (or the data) puts a value in the variable. Case only matches that value to a label.

4. A typical Grade 10 menu

On the form you show:

1 = Add 2 = Edit 3 = Delete 4 = Quit

The user types 2. The program stores 2 in iChoice. Case sees 2 and runs only the Edit line. It never also runs Add or Delete.

If they type 9, none of 1–4 match, so else says Invalid. That is why else is useful on a menu: people type wrong keys.

5. When to use If, when to use Case

Situation	Use	Why
Yes / no (stock low?)	If	Only two roads: true or false
A range (mark >= 50)	If / Else If	Case wants exact values, not “from 50 up”
One variable, list of exact options	Case	Shorter and easier to read than many Else Ifs
Menu 1, 2, 3, 4	Case	Each number is one exact choice

6. Small extra rules

- You may group values that share an action: 4, 5: ShowMessage('Reports');
- Two or more lines in one option need begin ... end.
- Case must end with end; — that end belongs to Case.
- The selector is usually an integer or a char ('A', 'B'). A long sentence in a string is still an If job in school Delphi.
- else is optional. Leave it out only if every possible value already has a branch.

7. One-line test reminder

If = one yes/no (or a range). Case = one variable matched against exact options. The user chooses by putting a value in that variable. Case just looks up which door that value opens.