

[7.37] Simulate SWITCH or CASE statement with GOTO and indirection

Some computer languages include a SWITCH or CASE statement which transfers program control to a label based on the value of an integer variable. For example, some versions of Basic implement this structure as an ON ... GOTO statement:

```
ON i GOTO I1,I2,I3
```

Program control is transferred to I1, I2 or I3, when $i=1, 2$ or 3, respectively.

TI-Basic does not have this statement built-in, but it can be simulated using *goto*, indirection and string manipulation.

First method

As an example, suppose we have a variable i and we want to branch to labels I1, I2 or I3 when i is 1, 2 or 3, respectively. The branch is accomplished with

```
goto #(mid("I1I2I3",2*i-1,2))
```

The program labels are specified as a single string, and the *mid()* function extracts the two label characters based on the value of i . Note that all the labels must have the same number of characters (two in this example), and i must be a valid value, or an error will result. The error can be avoided and handled by using the *try...endtry* structure.

In general, if each label is n characters long, then the general form of the expression is

```
goto #(mid(label_string,n*i-(n-1),n))
```

For example, if each label has four characters, then the expression is

```
goto #(mid(label_string,4*i-3,4))
```

2nd method

This method needs no string of all the labels, and directly builds the label from the index i . The general form is

```
goto #(label_root&string(exact(i)))
```

where *label_root* is the 'root' of the label names, and i is the index. For example, if the labels are I1, I2 and I3, then the root is "I" and the expression is

```
goto #("I"&string(exact(i)))
```

The *exact()* function is used to convert the index to an integer string with no decimal points; see tip [8.1] for details. As with the first method, *try...endtry* can be used to trap errors from invalid index values.

(Credit to Samuel Stearly (first method) and Stuart Dawson (second method))