

[7.35] Conditional test in *If ...EndIf* evaluates all 'and' arguments

Combining some arguments with the Boolean *and* operator will cause a TI Basic program to fail. For example:

```
If GetType(lista)="LIST" and lista[1]=2 then
  3→lista[1]
EndIf
```

In this example a program error occurs if the variable *lista* is not actually a list. This problem is avoided by using two nested *If* statements:

```
If GetType(lista)="LIST" then
  If lista[1]=2 then
    3→lista[1]
  EndIf
EndIf
```

{Behavior discovered by Kosh DV}