# Side Effects

Functions are supposed to come up with a single value. That is all they are supposed to do. If they do anything else (like changing the value of a global variable, either directly or through a variable parameter), this is known as a side effect. Just as a side effect of some medicine (like nausea, drowsiness, and death) is generally undesirable, a side effect in a computer program is usually undesirable as well. There are few exceptions although we'll see one in the next section. Here is an example of a program with two side effects.

|  |  |
| --- | --- |
| |  | | --- | | **var** number, x, cubed : **int**  **function** cube (**var** number : **int**) : **int**  x := number  number := number \* number  number := number \* x  **result** number  **end** cube  **put** "What is your favorite number?"  **get** x  **put** "What number would you like to cube?"  **get** number  cubed := cube(number)  **put** number, " cubed is ", cubed  **put** "Your favorite number is ", x | |

The two side effects are that it changes the value of the global variable x and it changes the variable parameter number. The function correctly calculates the cube of number, but it also changes the values of two other variables that cause the program to produce incorrect output. Here is an example of running it.

|  |
| --- |
| What is your favorite number?  **25**  What number would you like to cube?  **3**  27 cubed is 27  Your favorite number is 3 |

Here is another example. The function is to find the average of all the positive numbers in an array that it is passed. The problem is that it uses the global variable count to keep track of how many positive numbers there are. The main program was using that variable to keep track of the total number of elements in the array. This is a side effect. The function is only supposed to find the positive average but it also changes count.

|  |  |
| --- | --- |
| |  | | --- | | **var** count : **int**  **var** a : **array** 1 .. 100 of **int**  **function** posAverage (numbers : **array** 1 .. \* of **int**, maxNums : **int**) : **real**  **var** sum : **int** := 0    count := 0  **for** i : 1 .. maxNums  **if** numbers(i) > 0 **then**  count := count + 1  sum := sum + numbers(i)  **end** **if**  **end** **for**  **if** count = 0 **then**  **put** "No positive values, will give a result of -1."  **result** -1  **else**  **result** sum / count  **end** **if**  **end** posAverage  count := 1  **put** "To quit enter -99999"  **loop**  **put** "Enter value # ", count  **get** a(count)  **exit** **when** a(count) = -99999  count := count + 1  **end** **loop**  **put** "The average of the positive values is ", posAverage(a, count)  **put** "The original array had ", count, " elements in it" | |

Here is a sample run of the program. It gives the correct average of the positive values, but notice how the number of elements is incorrect.

|  |
| --- |
| To quit enter -99999  Enter value # 1  **10**  Enter value # 2  **-20**  Enter value # 3  **30**  Enter value # 4  **-40**  Enter value # 5  **50**  Enter value # 6  **-99999**  The average of the positive values is 30  The original array had 3 elements in it |

In general, you should try to avoid side effects. We've said before that we should almost never use global variables. That goes for both procedures and functions. Additionally, since functions are only supposed to do one thing, they should not usually use variable parameters.