Correspondence

The *cand* operator and its programming problems

In a letter to the editor [3], Wlodarczyk proposes a new programming construct called *andif* to solve the problem of nested *if*-statements in Pascal. Such a programming problem is indeed well-known to most experienced programmers, and is more generally related to the *and* operator. According to Jensen and Wirth [2] the *and* operator in Pascal may or may not be implemented as a *cand* operator—it is up to the compiler writer to decide, hence the confusing state of affairs. Thus instead of writing

```
if ⟨C1⟩ and ⟨C2⟩
        then ⟨A1⟩
        else ⟨A2⟩
```

one is forced to write

```
if ⟨C1⟩
       then if ⟨C2⟩
              then ⟨A1⟩
              else ⟨A2⟩
        else ⟨A2⟩
```

just to be sure that the effect of the *cand* is achieved. However, if ⟨A2⟩ is a lengthy programming code, the unnecessary duplication of it is not desirable. Fortunately, the termination-indicator technique [1] can be used to resolve this tricky programming program. To make the discussion concrete, we now apply it to the zero-denominator (or divided-by-zero) problem. Suppose the original intended programming solution is

```
if D≠0 cand N/D>1
        then ⟨A1⟩
        else ⟨A2⟩.
```

Applying the termination-indicator technique, we obtain

```
t:=false; {termination indicator}
if D≠0 then t:=N/D>1;
if t
        then ⟨A1⟩
        else ⟨A2⟩.
```

With this, the proposed *andif* construct is no longer required.

M. Er
ICS Dept, KFUPM, Box 1779, Dhahran 31261, Saudi Arabia.

ACKNOWLEDGMENT

The use of KFUPM's facilities is acknowledged by the author.

REFERENCES

[1] M. C. Er, The use of termination indicators in computer programming, *The Computer Journal* **29**, pp. 430–433 (1986).
[2] K. Jensen and N. Wirth, *PASCAL — User Manual and Report*, Springer-Verlag, New York (1978).
[3] A. J. Wlodarczyk, Letter to the Editor. *The Computer-Journal* **32**, pp. 380–381 (1989).