1 Problems

1. Argue that if a programming language has a while statement, it does not need an if statement.

**Solution:** The principal difference between an if construct and a while construct is that if a condition is satisfied, the former executes a statement once, whereas the latter could execute it multiple times. In order to enforce that the while construct executes a statement precisely once, when a condition is satisfied, we use the following strategy: Let cond denote the condition for entering the if construct. Use (cond ∧ flag) as the condition for entering the while construct, where flag is set to true, before entering the loop and set to false, immediately upon entry. This will guarantee that the statement inside the while construct is executed precisely once, thereby mimicking the if construct.

2. Write a Prolog-style fragment to implement the factorial function.

**Solution:**

1. fact(U, 1) : - U = 0.
2. fact(U, V) : - not (U = 0), fact(U - 1, Y), V is U * Y.

*Algorithm 1.1:* Implementing the factorial function in Prolog

3. Write a functional-style (i.e., C with recursion) fragment for the problem of determining the number of digits of a positive integer.

**Solution:**

```
Function NUMDIGITS( n )
1: if ( n / 10 == 0) then
2: return( 1 )
3: else
4: return( 1+NUMDIGITS(n / 10) )
5: end if
```

*Algorithm 1.2:* A functional algorithm for counting digits of a positive number
4. Discuss how the following features have been promoted and violated in the C programming language: (a) Generality, and (b) Extensibility.

**Solution:**

(a) Generality in C, has been promoted by the existence of variable-length arrays and violated by the non-existence of direct comparison between structures.

(b) Extensibility in C, has been promoted by the overloading of the “+” operator for addition in that the programmer does not have to specify whether he is adding floats or integers. Extensibility has been violated to the extent that the “+” operator cannot be overloaded for matrix addition.

☐