1. Consider the following Modula-2 program that calculates the greatest common divisor of two integers $x$ and $y$:

Note the following: every Modula-2 program has a name. The FROM and IMPORT here are just to include standard input and output. A procedure in Modula-2 followed by a ";" then data type is a function that returns a value (vs. the usual void function in C). Note that this function is recursive. Note also that MOD has the same meaning as "\%" for non-negative integers.

```modula2
MODULE ModSample;

FROM InOut IMPORT WriteString, ReadInt, WriteInt, WriteLn;

PROCEDURE gcd( u, v : INTEGER ) : INTEGER;
BEGIN
  IF v = 0 THEN
    RETURN u
  ELSE
    RETURN gcd( v, u MOD v )
  END;
END gcd;

VAR x, y : INTEGER;
BEGIN (* main program *)
  WriteString( 'Input two integers: ' );
  WriteLn;
  ReadInt(x);
  Writeln;
  ReadInt(y);
  WriteLn;
  WriteString( 'The gcd of ' );
  WriteInt( x, 1 );
  WritelnString( ' and ' );
  WriteInt( y, 1 );
  WriteString( ' is ' );
  WriteInt( gcd( x, y ), 1 );
  WriteLn;
END ModSample.
```

Translate this program into a C++ program here:
2. Consider the following Pascal function that returns the number of digits in an integer:

Note here that div is ordinary integer division. As in Modula := is the assignment. div is integer division like “/” in C++ between two integers.

```pascal
function numdigits (x: integer): integer;
var
  t, n: integer;
  n := 1;
  t := x;
  while t >= 10 do begin
    n := n + 1;
    t := t div 10
  end;
numdigits := n
end;
```

**Translate this function into a C++ function here:**

3. What can you say about the following?

a. How are semicolons used in Pascal differently than in C++?

b. The begin and end statements in Pascal and Modula-2 correspond (almost exactly) to what in C++?

c. The way of returning a value in a Modula-2 procedure is very similar to a value-returned function C++. However, how does Pascal return a value in a function?