(1) The procedure multiply in page 23 can be explained by using the following hand calculation.

```
1 1 1     a=7
1 0 1 0 1 0 1     b=85
```

```
1 1 1     a * 1
0 0 0     2 a * 0
1 1 1     2 a * 1
0 0 0     2 a * 0
1 1 1     2 a * 1
0 0 0     2 a * 0
1 1 1     2 a * 1
1 0 0 1 0 1 0 0 1 1     595
```

The computation of while loop proceeds from top to bottom in this figure. In each iteration in the loop, a is doubled and b is halved. The least significant bit of b is tested. If it is 1, a is added to z. Explain procedure divide in the same page in a similar way.

(2) The following program multiply two numbers with addition and subtraction.

```
{x>=0}
  a:=x; b:=y; z:=0;
  while a>0 do begin
    z:=z+b; a:=a-1
  end. {z=x*y}
```

Which multiply is more efficient? Give the reason.

(3) The following program divides x by y.

```
{y>0}
  r:=x; q:=0; w:=y;
  while w <= r do begin
    r:=r-w; q:=q+1
  end. {x=q*y+r and 0 <= r < y, q is quotient, r is remainder}
```

Which divide is more efficient? Give the reason.

(4) Trace procedure gcd in page 23 with x=84 and y=35.

(5) The procedure gcd in page 23 can be rewritten as follows:
procedure gcd;
  var f, g, h;
begin
  f:=x; g:=y;
  while g > 0 do begin
    h:=f mod g;
    f:=g;
    g:=h
  end
end.

Trace this procedure with the same x and y.

(6) This procedure is further modified into

procedure gcd;
  var f : history of degree 1;                 f has two words; one for the current value
begin                                                    the other for the previous value. If a new
  f:=x; f:=y;                                           value is assigned with f, the current value
  while f > 0 do f:=f<1> mod f              is moved to f<1> automatically.
  end.
  { f<1> = gcd(x, y) }

Trace this procedure with the same x and y.

(7) Enhance the syntax chart of PL/0 with array declaration like

    var a : array[lower .. upper];

where lower and upper are constants and array reference like a[i], a[i+j], etc.

(8) Enhance the chart with repeat statement.

(9) Enhance the chart with go to statement.

(10) Enhance the chart with if-then-else statement.
If you enhance the chart in this question, there will be ambiguity like

        if B1 then if B2 then S1 else S2
        if B1 then (if B2 then S1 else S2)
        if B1 then (if B2 then S1 else S2)

Semantically these two interpretations cause different results. Give such an example.

(11) Trace the object code in page 36 with x=3 and y=5. Trace the changes on the stack s.