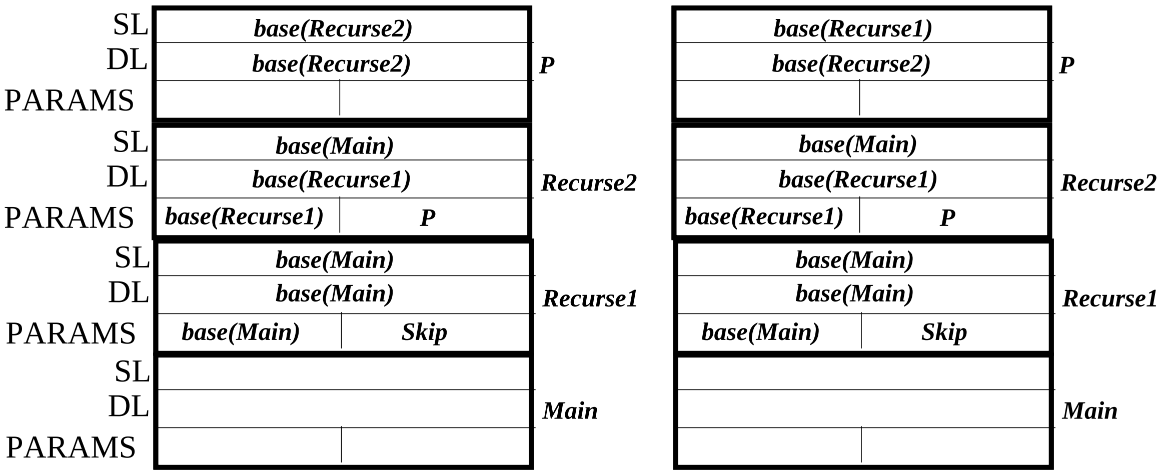
Max. Marks 20 **QUIZ 3** Date: 18th March 2002

1. Draw the stack of activation records every time control reaches the point marked `-->` during execution of the program shown below. In an activation record show (a) the dynamic link, (b) the static link, and (c) the parameter values. **(8 Marks)**

```
program Main (input, output);
     procedure Recurse (i:integer; procedure Q);
         procedure P;
            begin
 -->          write(i)
            end
         begin {Recurse}
            if i > 0 then  Recurse (i-1, P)
            else begin
                    P; Q
                 end
         end {Recurse}
     procedure Skip;
         begin  end;
    begin {Main}
       Recurse(1,skip)
    end.
```




***P executed directly*** ***P called through parameter Q***

2. Generate code for the `gcd` function which will use the instructions `LINK`, `UNLK`, `JSR`, `RSR` and `PEA`. Any other instructions like `MOVE` can be expressed in terms of assignments. Addressing modes like indexing and indirection can be expressed using + and C-style `*`, and loops can be expressed using `if` with `goto`s. Assume that the underlying machine directly supports the instruction `mod`.

```
function gcd (var a, b: integer): integer;
    var c: integer;
  begin
    if a=0 then gcd := b
    else
      begin
        c := b mod a;
        gcd := gcd(c,a)
      end
  end
```

Also, to aid correction, draw a picture showing what goes into different slots in the activation record of gcd. **(12 Marks)**

```
      gcd:  LINK BP, #-1

            if **(BP+4)== 0 go to then     /* if a = 0 go to then*/
     else:  R1 := *(BP+3)                  /* else c = b mod a*/
            R2 := **(BP+4)
            R1 := R1 mod R2
            *(BP-1) := R1

            SP := SP-1                     /* ret. value */
            PEA   (BP-1)         /* push address of c, actual for a */
            push **(BP+4)        /* push value of a, actual for b */

            R1 := BP                   /* set up static link */
            R2 := 1
     loop:  if R2 == 0 go to end_loop
            R1 := *(R1+2)
            R2 := R2 - 1
            go to loop
end_loop:   push R1                    /* static link set */

            JSR gcd

            SP := SP+4     /* clean up act. rec. of callee */
            go to end

     then:  *(BP+5)= *(BP+3)          /* then gcd := b */
      end:  UNLK BP
            RSR
```