Ques no. 01 GATE

To fix the error, we need to ensure that the result of the division in procedure `P` is assigned as an integer since `x` is declared as an integer. One way to do this is to use integer division (with the `div` operator) instead of real division (`/`).

Here's the corrected program:

```
pascal
var
  x, y : integer;

procedure P(n : integer);
begin
  x := (n + 2) div (n - 3);  // using 'div' for integer division
end;

procedure Q;
var
  x, y : integer;
begin
  x := 3;
  y := 4;
  P(y);
  Write(x);  // Local x is written here
end;
```

```
begin
 x := 7;
 y := 8;
 Q;
 Write(x);  // Global x is written here
end.
```

Explanation of the fix:

- We changed the division operator `/` to `div`, which performs integer division in Pascal. This ensures that the result is an integer.

- Since `n - 3` in `P(n)` can potentially be zero if `n` is 3, we'd need to ensure that `n` is not 3 to avoid a division by zero. However, in this case, `n = 4`, so no division by zero occurs.

Step-by-Step Output Calculation:

1. Global Variables:
   - Initially, `x = 7` and `y = 8`.

2. Call to Procedure `Q`:
   - Local variables `x` and `y` are declared within `Q`.
   - Local `x` is assigned 3, and local `y` is assigned 4.
   - `P(y)` is called, passing the local `y = 4` to procedure `P`.

3. Call to Procedure `P`:
   - `P(4)` is called, so `n = 4`.
   - The expression in `P` becomes `x := (4 + 2) div (4 - 3)` → `x := 6 div 1` → `x := 6`.
   - This updates the global variable `x` to 6, since `P` modifies the global `x`.

4. Return to Procedure `Q`:

- The `Write(x)` statement inside `Q` prints the local `x`, which is 3.

5. After Procedure `Q`:

- The program returns to the main body, and the `Write(x)` statement prints the global `x`, which was updated to 6 by the call to `P`.

Output:

3

6

- The first `Write` statement (inside `Q`) prints the local `x = 3`.

- The second `Write` statement (after `Q` in the main program) prints the global `x = 6`.