1. CS5391 Survey of Software Engineering

   (a) state the pros and cons of lines of code and function points metrics
   (b) describe incremental development model and its limitation

2. CS5392 Formal Methods in Software Engineering

   Provide a partial correctness proof for the following function within a precondition and a postcondition (show all the intermediate assertions including an invariant and the inference rules applied at each step). The function computes \( z = \lfloor \sqrt{x} \rfloor \) for every natural number \( x \); that is the final value of \( z \) is the largest integer \( k \) such that \( k \leq \sqrt{x} \). The computation is based on the fact that \( 1 + 3 + 5 + \ldots + (2n + 1) = (n + 1)^2 \) for every \( n \geq 0 \). \( n \) is in \( y_1 \), \( (2n + 1) \) is in \( y_3 \), and the sum \( 1 + 3 + 5 + \ldots + (2n + 1) \) in \( y_2 \).

   \[ \{ x \geq 0 \} \]

   \[
   \begin{align*}
   y_1 &= 0; \\
   y_2 &= 0; \\
   y_3 &= 1; \\
   y_2 &= y_2 + y_3; \\
   \text{while}( y_2 <= x ) \{} \\
   y_1 &= y_1 + 1; \\
   y_3 &= y_3 + 2; \\
   y_2 &= y_2 + y_3; \\
   \} \\
   z &= y_1; \\
   \{ \lfloor z^2 \rfloor \leq x < (z+1)^2 \}
   \end{align*}
   \]
3. CS5393 Software Quality

Refer to the pseudo-code that follows, construct a corresponding Control Flow Graph and show all work to accomplish the following: List the smallest set of paths required to achieve all-uses coverage and the paths themselves.

Notes: The All-uses criterion requires definition-clear paths from each definition to each use reached by that definition and each successor node of the use. The paths must be from the start node to the end node of your CFG.

```plaintext
1. start
2. read p, e
3. d:=1
4. x:=0
5. c:=2*p
6. if c >= 2 then goto 18
7. if d <= e then goto 16
8. d:=d/2
9. t:=c-(2*x+d)
10. if t<0 then goto 14
11. x:=x+d
12. c:=2*(c-(2*x+d))
13. goto 7
14. c:=2*c
15. goto 7
16. print x
17. stop
18. print "error"
19. stop
```