1. Scope_Diagram.png ---> it contains scope diagram asked in Question 1, part (i)

    Note: In Scope_Diagram.png , 'ioG' stands for 'inorder_gen'

    Explanation:

a. main frame is created
   - a. ioG^1 frame is created
     - i. ioG^2 frame is created
       1. ioG^3 frame is created, and tr is None, so control is returned back to ioG^2
     - ii. $thk2^1$ frame is created
       1. thunk frame is created and prints the value 10, then control is returned back to $thk2^1$
       2. $thk2^1$ returns the control to ioG^2
     - iii. ioG^2 returns the control to ioG^1
   - b. $thunk^2$ frame is created inside ioG^2
     - i. prints the value 20
   - c. ioG^4 frame is created
     - i. ioG^5 frame is created
       1. ioG^6 frame is created and tr is None, so control is returned back to ioG^5
     - ii. $thk2^2$ frame is created
       1. $thunk^3$ frame is created, prints the value 30, and returns the control to $thk2^2$
       2. $thk2^2$ returns the control to ioG^5
     - iii. ioG^5 returns the control to ioG^4
   - d. $thk2^3$ frame is created inside ioG^4
     - i. $thunk^4$ frame is created, prints the value 40
     - ii. $thunk^4$ returns the control to $thk2^3$
   - e. $thk2^3$ returns the control to ioG^4
   - f. ioG^4 returns the control back to ioG^1
b. ioG^1 returns the control to main

2. Stack_Diagram.png ---> it contains stack diagram asked in Question 2, part (ii)

3. flatten.py ---> contains solution of Question 2

4. dfirst2.sml ---> contains solution for Question 3

4. ntree.sml ----> contains solution for Question 4