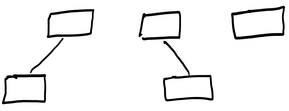
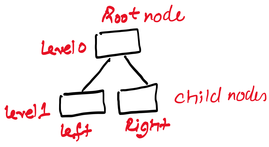
Binary Tree (BT)

Tuesday, 10 November 2020 8:11 AM

Binary

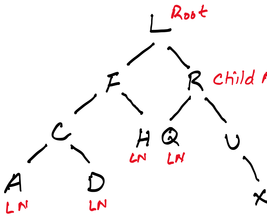
2





Insert

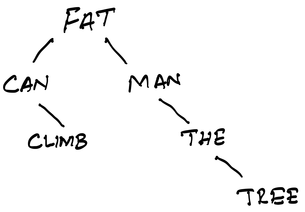
L, F, R, C, U, A, X, H, D, Q



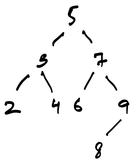
Insert

```
INPUT DATA
GOTO Root
IF Root = Null
  THEN
    CREATE NODE
    INSERT DATA
    END
ENDIF
WHILE (Node <> Null)
  COMPARE WITH NODE
  IF LESS THEN GOTO LEFT
  IF GREATER THAN GOTO RIGHT
ENDWHILE
CREATE NODE
ENTER DATA
END
```

FAT MAN CAN CLIMB THE TREE.

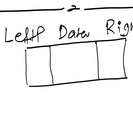


5, 3, 7, 9, 2, 6, 8, 4



Searching BT.

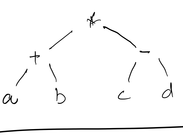
```
INPUT DATA.
IsFound ← FALSE
GOTO ROOT
IF ROOT = Null
  THEN
    OUTPUT "NOT FOUND"
  ELSE
    WHILE(NODE.DATA
          <> DATA
            AND
          IsFound = FALSE
          AND NODE <> NULL)

      COMPARE DATA
        WITH Node.DATA
      IF LESS GOTO LEFT
      IF GREATER GOTO RIGHT

      IF Node.DATA = DATA
        THEN
          IsFound = TRUE
      ENDIF
    ENDWHILE
    IF IsFound = TRUE
      THEN
        OUTPUT "FOUND"
      ELSE
        OUTPUT "NOT FOUND"
    ENDIF
```



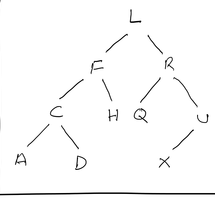

```
TYPE binNode
    DECLARE LP : INT
    DECLARE Data : STR
    DECLARE RP : INT
ENDTYPE
```

Maths Expressions

(a+b) * (c-d)



L, F, R, C, U, A, X, H, D, Q




```
//SETUP binTree
RECORD binTree
    DECLARE LeftP : INTEGER
    DECLARE Data : CHARACTER
    DECLARE RightP : INTEGER
END RECORD

DECLARE BT : ARRAY[0:9] OF binTree

FOR i ← 0 TO 9
    BT[i].LeftP ← -1
    BT[i].RightP ← -1
    BT[i].Data ← ''
NEXT i
CONSTANT Root ← 0
```

	LeftP	Data	RightP
0	1	L	2
1	3	F	7
2	9	R	4
3	5	C	8
4	6	U	-1
5	-1	A	-1
6	-1	X	-1
7	-1	H	-1
8	-1	D	-1
9	-1	Q	-1

BT