Method : Some Methods to store a Password, a File, a Database using a tree within a File OR within a Directory structure on a File system.

```
Procedure_A()
{

    Create a Tree with labeled nodes called TreeHashedPassword.
    Apply Key to password to Obtain HashedPassword.
    Partition the HashedPassword into variable length partitions of bits
        (Partiton1, Partition2..PartitionN).
    Create a random sequence within the label space of labeled nodes in tree (P1,P2,P3...PN)
    Store the random sequence in RS0[];
    Assign each Pi to Partition(i).
    Store the Partition(i) in Pi using insert function into TreeHashedPassword.

}

Procedure_B()
{

    Create a Tree with labeled nodes called TreeKeyPartitions
    Next Partition the Key into M partitions (Partition1, Partition2, ... PartitionM)
    Create a random sequence within the label space of labeled nodes in
        TreeKeyPartitions (K1,K2...K(M))
    Store the random sequence in RS1[];
    Assign each K(i) to Partition(j) where j = 1..M.
    Store the Partition(j) in K(i) using insert function into TreeKeyPartitions.

}

Procedure_Reconstruct_Password()
{
    Use the above routines to fetch based on RS0[], RS1[].
    Reconstruct the HashedPassword from RS0.
    Reconstruct the Key from RS1.
    Unhash the HashedPassword using Key.
    Use password to login.
    Hash Password with new Key.
}

Key_init_or_rollover()
{
    Key rollover involves resorting to executing all procedures (A),(B).
}
```

303 307 LEFT lev=13 ***->
289 303 LEFT lev=12 ***->two find this
303 309 RIGHT lev=13 ***->
285 289 LEFT lev=11 ***->
289 305 RIGHT lev=12 ***->fourfind this
281 285 LEFT lev=10 ***->
285 301 RIGHT lev=11 ***->
295 281 LEFT lev=9 ***->
281 287 RIGHT lev=10 ***->
291 295 LEFT lev=8 ***->
295 283 RIGHT lev=9 ***->
29 291 LEFT lev=7 ***->
291 297 RIGHT lev=8 ***->
25 29 LEFT lev=6 ***->
29 293 RIGHT lev=7 ***->
21 25 LEFT lev=5 ***->
25 32 RIGHT lev=6 ***->
16 21 LEFT lev=4 ***->
21 27 RIGHT lev=5 ***->
12 16 LEFT lev=3 ***->
16 23 RIGHT lev=4 ***->
10 12 LEFT lev=2 ***->
12 18 RIGHT lev=3 ***->
100 10 LEFT lev=1 ***->
10 14 RIGHT lev=2 ***->
100 100 ROOT lev=0
11 15 LEFT lev=2 ***->
100 11 RIGHT lev=1 ***->
13 19 LEFT lev=3 ***->
11 13 RIGHT lev=2 ***->
17 24 LEFT lev=4 ***->
13 17 RIGHT lev=3 ***->
22 28 LEFT lev=5 ***->
17 22 RIGHT lev=4 ***->zerofind this
26 33 LEFT lev=6 ***->
22 26 RIGHT lev=5 ***->
31 294 LEFT lev=7 ***->
26 31 RIGHT lev=6 ***->
292 299 LEFT lev=8 ***->
31 292 RIGHT lev=7 ***->
296 284 LEFT lev=9 ***->
292 296 RIGHT lev=8 ***->
282 288 LEFT lev=10 ***->
296 282 RIGHT lev=9 ***->
286 302 LEFT lev=11 ***->one find this
282 286 RIGHT lev=10 ***->
300 306 LEFT lev=12 ***->
286 300 RIGHT lev=11 ***->
300 304 RIGHT lev=12 ***->threfind this
304 308 RIGHT lev=13 ***->