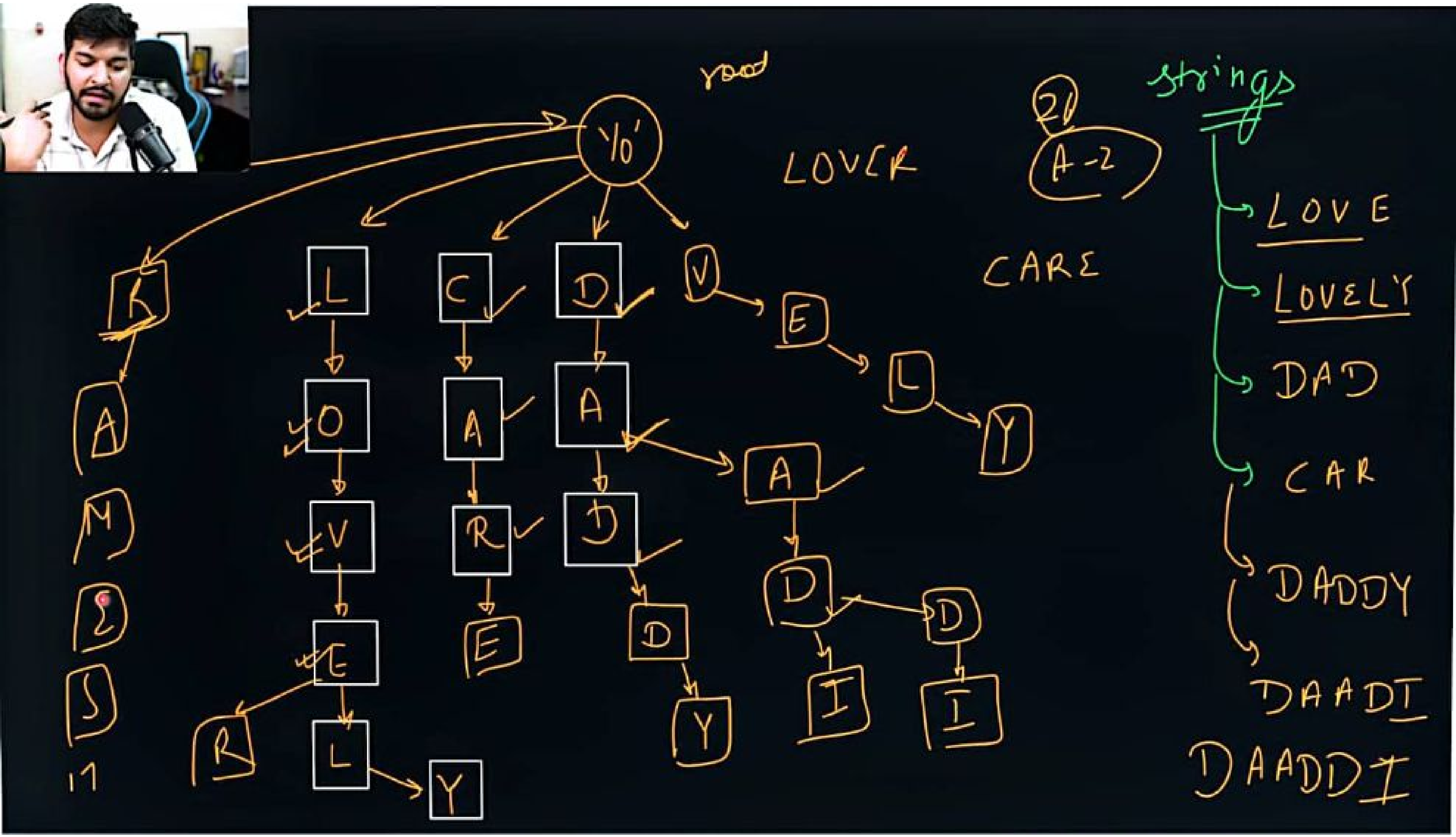
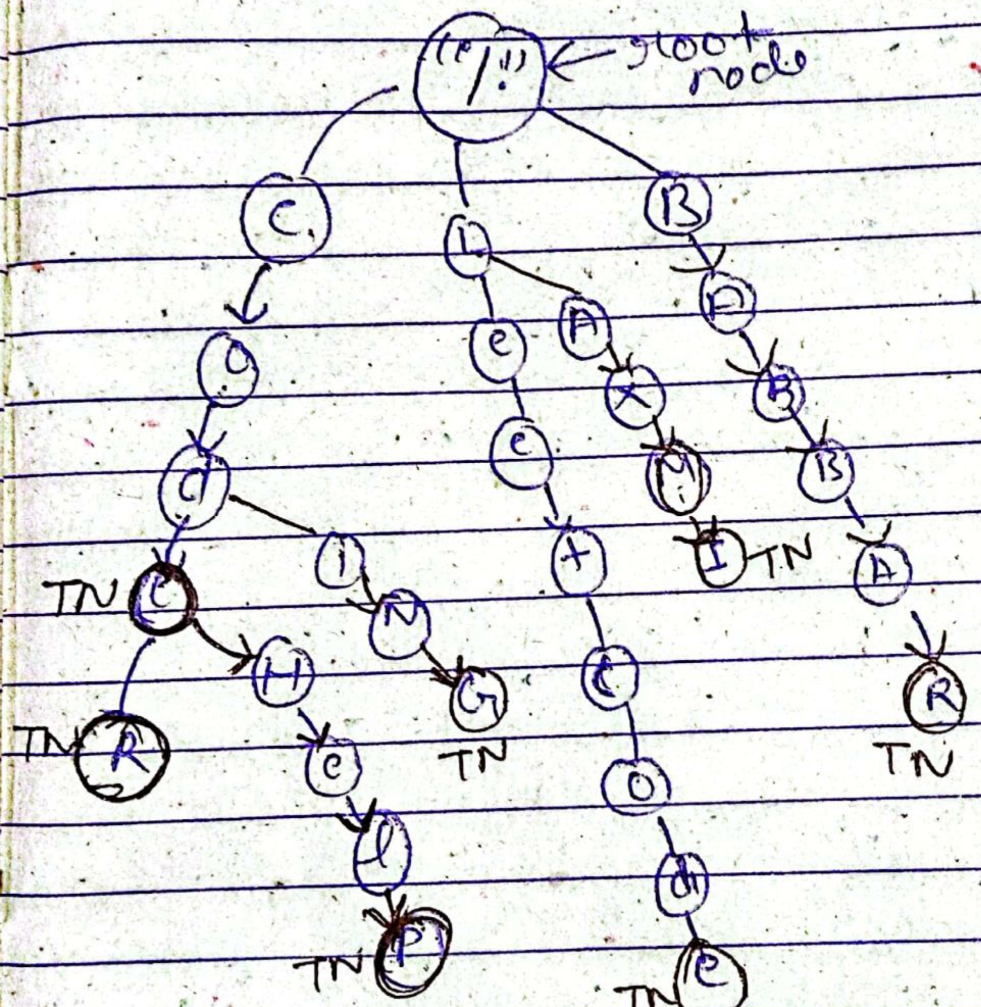


root
'\0'
LOVER
26
A-Z
strings
LOVE
LOVELY
DAD
CAR
DADDY
DAADI
DAADDI
CARE
K A M E S
17
L O V E R L Y
C A R E
D A D D Y
A D I D I
V E L Y

Trie → is a type of DataStructure

- → multi-way tree structure.
- → Generally used for Pattern Searching



String

- ✓ Code
- ✓ Coder
- ✓ Codehelp
- ✓ Coding
- ✓ Leetcode
- ✓ Laxmi
- ~~Calling~~
- ✓ Babbar
- ~~India~~
- ~~Supreme~~

If we want to delete Code string so we mark terminal node as normal node.

We can say Any string is present or not, if string ka last char terminal node mark hai.

Insertion

- → node present
  - ↳ node pr chale jao.
- → node absent
  - ↳ node create kro & node pr chale jao

$TC \rightarrow O(\underset{\substack{\uparrow \\ \underline{\text{word length}}}}{L})$

Longest Common Prefix

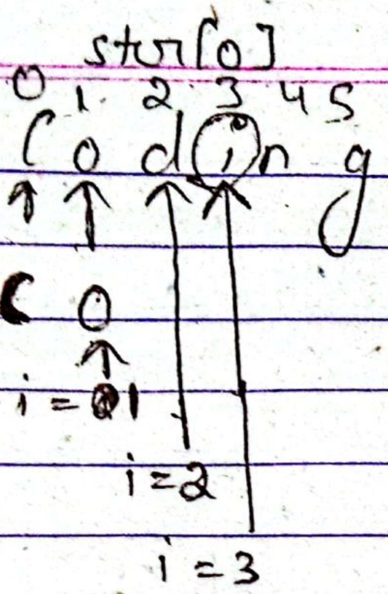
strs = ["Coding", "Codex", "Coder"];

isko reference bnakr baaki sb string se compare krenge,
agr isko ref- se ~~str~~ 0 to element milega, usko string me daal denge

```
for ( int i=0; i<str[0].size; i++ )
{
    ch = str[0][i]
    match = true;

    for ( j=1 ; j <str, size(); j++ )
    {
        if str[i] <i || ch != str[j][i]
        ) {
            match = false
            break)
        }

    if (match ==f) break
else
    ans.push back(ch);
```



0 1 2 3 4
Codix

i = 1

str[j][i]
str[1][0]
✓ C

str[j][i]
str[1][1]
o

str[j][2]
str[1][2]
✓ d

str[j][3]
str[1][3]
e ✗

0 1 2 3 4
Coder

j = 2

str[j][0]
str[2][0]
✓ C

str[j][i]
str[2][1]
✓ o

str[j][2]
str[2][2]
✓ d

str[j][3]
str[2][3]
e ✗

~~str = C~~

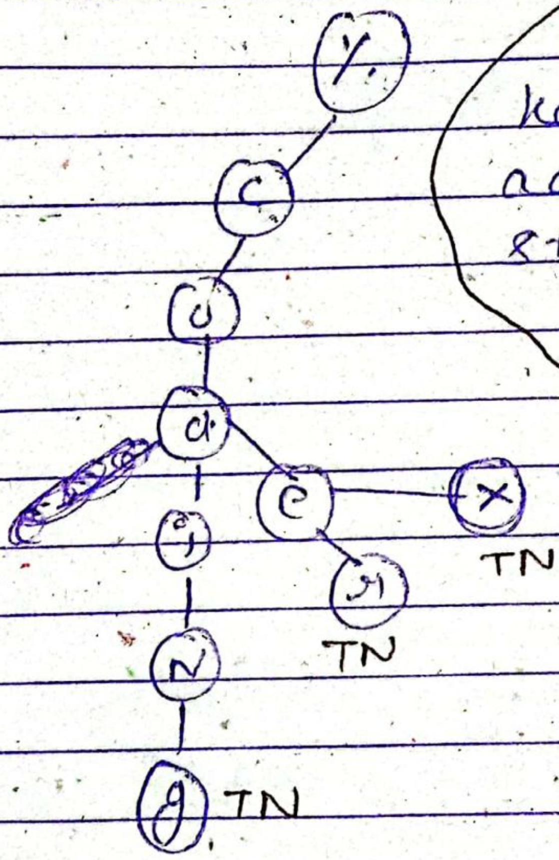
break
i = 3
str[0][3]
i ✗

$$\boxed{\text{Str = cod}}$$

2nd Approach (using Trie)

Make Trie Data Tree like Structure

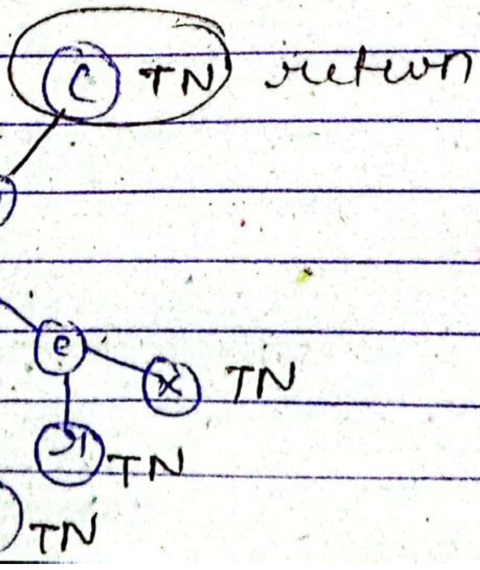
str = ["coding", "codex", "coder"]

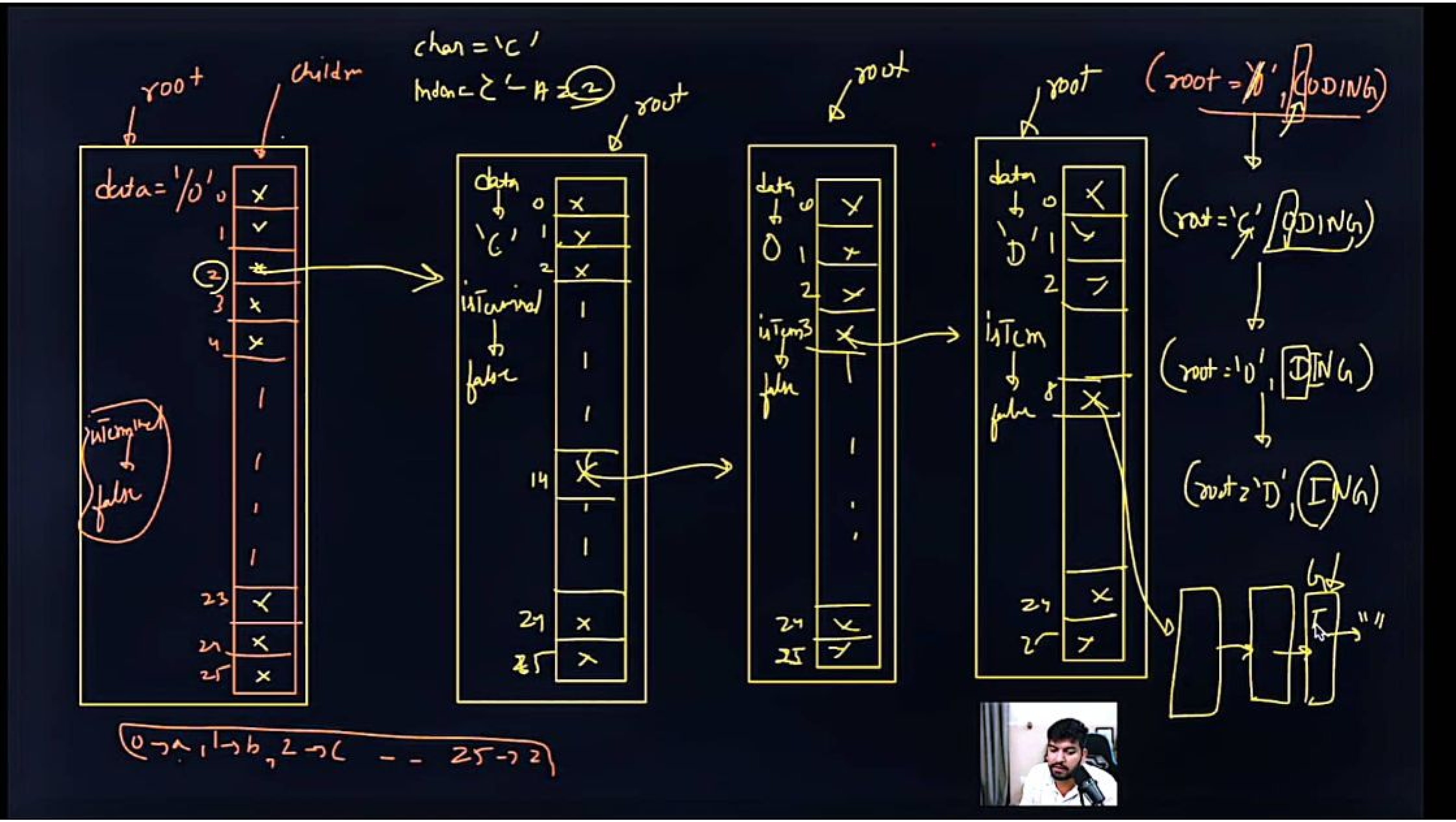


jo jis ik node ka child count == 1 add this into string ans;.

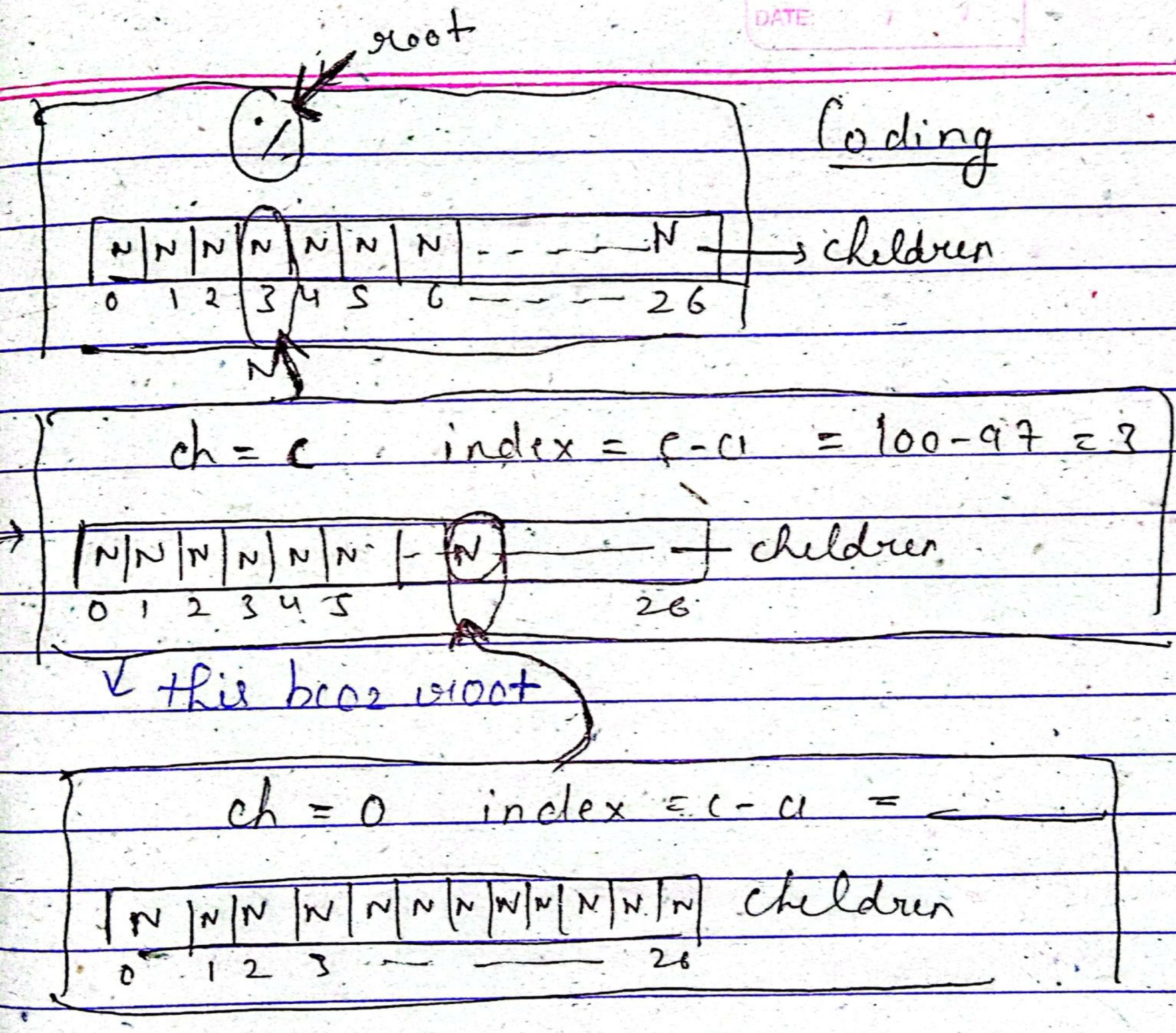
if (koi node terminal) so return

["coding", "codex", "c", "coder"]



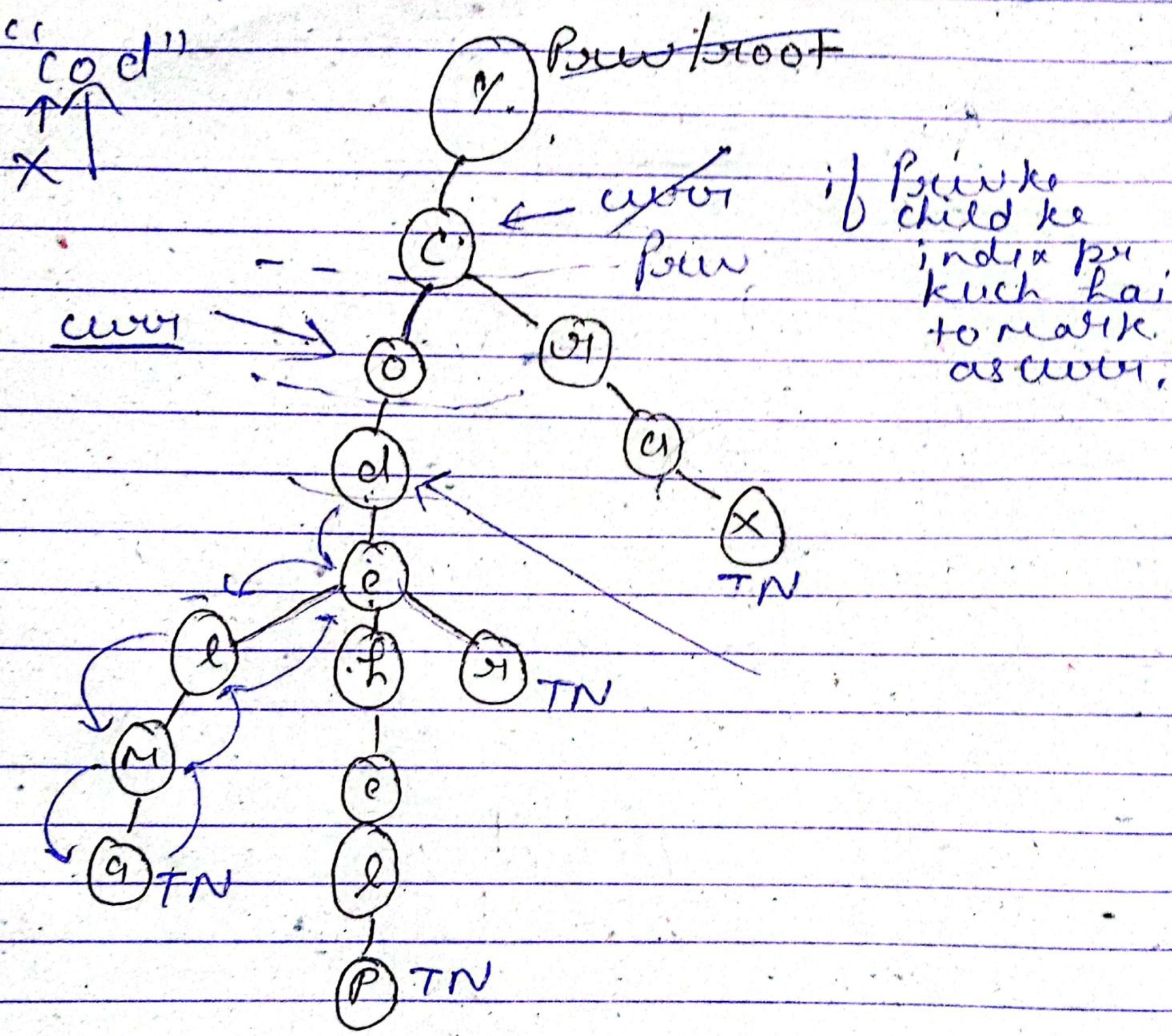

char = 'C'
Index = 'C' - 'A' = 2
root
children
data = '\0'
isTerminal → false
data → 'C'
isTerminal → false
data → 'O'
isTerm → false
data → 'D'
isTerm → false
root
(root = '\0', CODING)
(root = 'C', ODING)
(root = 'O', DING)
(root = 'D', ING)
0 → a, 1 → b, 2 → c - - 25 → z

PAGE NO.
DATE: / /



root
/
N N N N N N N ---- N
0 1 2 3 4 5 6 ---- 26
children
Coding
N
ch = c index = c - a = 100 - 97 = 3
N N N N N N - N
0 1 2 3 4 5 26
children
this bcoz root
ch = o index = c - a = ___
N N N N N N N N N N N N
0 1 2 3 ___ ___ 26
children

vector → [codehelp , coder , codelma]
oeax



c ko give choice a to z
jaha tk wo jaa skta hai.