

Day - 51Strings Interview Questions* Defanging an IP address:

= You will get an address like this —

address = 1.1.1.1

= We have to write this in the form —

ans = 1[.]1[.]1[.]1

Ex: address = 255.100.25.60

255[.]100[.]25[.]60

(click)

= We will check if '.' come then we will add square bracket ([.]) around it.

Code

```

int index = 0;
String ans;
while (index < address.size()) {
    if (address[index] == '.') {
        ans = ans + "[.]";
    } else {
        ans = ans + address[index];
    }
    index++;
}
return ans;

```

T.C. → O(N)

S.C. → O(N)

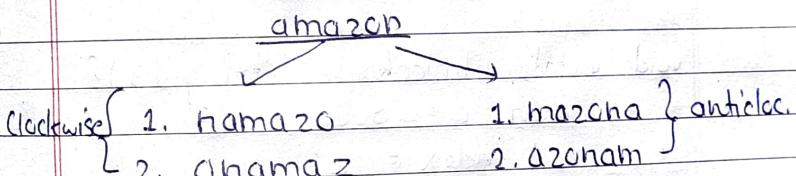
* Check if string is rotated by 2 Places

= We have two strings, we have to check that if ~~=~~ we rotated str1 by 2 places then we will get str2 or not.

str1 = amazon

str2 = azonam

=



Code

str1, str2

String clockwise = str1, anticlockwise = str1;

rotate cw (clockwise);

rotate cw (clockwise);

if (clockwise == str2)

 return 1;

rotate acw (anticlockwise);

rotate acw (anticlockwise);

if (anticlockwise == str2)

 return 1;

return 0;

```

void rotateclockwise ( string &s ) {
    char c = s[ s.size() - 1 ];
    int index = s.size() - 2;
    while ( index >= 0 ) {
        s[ index + 1 ] = s[ index ];
        index--;
    }
    s[ 0 ] = c;
}

```

```

void rotateanticlockwise( string &s ) {
    char c = s[ 0 ];
    int index = 1;
    while ( index < s.size() ) {
        s[ index - 1 ] = s[ index ];
        index++;
    }
    s[ s.size() - 1 ] = c;
}

```

* Check Pangram:

→ We have to check the given sentence contains all the alphabets or not.

= sentence = "the quick brown fox jumps over the lazy dog."

Date _____

Page _____

- =1 we will create an array of 26 size and then iterate the sentence and we will increase the count of alphabets,

$$a \rightarrow 0$$

$$b \rightarrow 1$$

$$\vdots \vdots$$

- =1 After that, if we get any zero in the array then return 0 otherwise the string is pangram.

$$\text{index} = \text{alpha} - 'a'$$

Code

```
vector<bool> alpha(26, 0);
for(int i=0; i < sentence.size(); i++) {
    int index = sentence[i] - 'a';
    alpha[index] = 1;
```

}

```
for(i=0; i < 26; i++) {
    if(alpha[i] == 0)
        return 0;
```

}

return 1;

$$\text{T.C} \rightarrow O(N) + O(26) = O(N)$$

$$\text{S.C.} \rightarrow O(1)$$

* Sort a string:

$$S = edcab$$

$$\text{ans} = \text{abcde}$$

\Rightarrow we can solve it in $O(N^2)$ time by using sorting alg.

=1 For solving it in $O(N)$, we can first count the alphabets after that arrange the alphabets in ascending order.

Code

```
vector<int> alpha(26, 0);
```

```
for( int i= 0 ; i < size(1) ; i++ ) {
```

int index = s[i] - '0';

alpha[index]++;

? string ans;

```
for( int i= 0; i< 26; i+){
```

```
char c = 'a'+ i;
```

while (alpha[i]) {

ans + = c;

alpha[i]--;

3

3

returnans;