

# Remove Element in String - Quick Revision Notes

**\*\*Logic of the Given Code:\*\***

- Function `removecharacter(string ans, string original, int idx)` removes all occurrences of 'a' from a string using recursion.
- **\*\*Base Condition\*\***: If `idx == original.length()`, the function prints `ans` and stops further recursive calls.
- **\*\*Recursive Calls\*\***
  - If the current character is 'a', it is skipped.
  - Otherwise, it is added to `ans`, and the recursion continues.

**\*\*Code with Comments:\*\***

```
```cpp
#include<iostream>
using namespace std;

// Function to remove 'a' from a string using recursion
void removecharacter(string ans, string original, int idx) {
    // Base condition: If index reaches end of string, print the result
    if(idx == original.length()) {
        cout << ans;
        return;
    }

    char ch = original[idx];
    // If current character is 'a', skip it
    if(ch == 'a') {
        return removecharacter(ans, original, idx + 1);
    }
    // Otherwise, include it in ans and proceed
    else {
        removecharacter(ans + ch, original, idx + 1);
    }
}
```

```
}  
}
```

```
int main() {  
    string str = "Raghav Garg";  
    removecharacter("", str, 0);  
}  
...
```

**\*\*Dry Run of the Code (For "Raghav Garg")\*\***

Function Call	`idx`	`ch`	`ans` before call	Action
removecharacter("", "Raghav Garg", 0)	0	'R'	""	'R' added to `ans`
removecharacter("R", "Raghav Garg", 1)	1	'a'	"R"	'a' skipped
removecharacter("R", "Raghav Garg", 2)	2	'g'	"R"	'g' added to `ans`
removecharacter("Rg", "Raghav Garg", 3)	3	'h'	"Rg"	'h' added to `ans`
removecharacter("Rgh", "Raghav Garg", 4)	4	'a'	"Rgh"	'a' skipped
removecharacter("Rgh", "Raghav Garg", 5)	5	'v'	"Rgh"	'v' added to `ans`
removecharacter("Rghv", "Raghav Garg", 6)	6	' '	"Rghv"	Space added
removecharacter("Rghv ", "Raghav Garg", 7)	7	'G'	"Rghv "	'G' added to `ans`
removecharacter("Rghv G", "Raghav Garg", 8)	8	'a'	"Rghv G"	'a' skipped
removecharacter("Rghv G", "Raghav Garg", 9)	9	'r'	"Rghv G"	'r' added to `ans`
removecharacter("Rghv Gr", "Raghav Garg", 10)	10	'g'	"Rghv Gr"	'g' added to `ans`
removecharacter("Rghv Grg", "Raghav Garg", 11)	11	'\0'	"Rghv Grg"	Print and return

**\*\*Final Output:\*\***

...

Rghv Grg

...

**\*\*Time Complexity:\*\***

- The function processes each character once, leading to a time complexity of **\*\*O(n)\*\*** where `n` is

the length of the string.

**\*\*Key Takeaways.\*\***

- Recursion helps process strings character by character.
- Base condition ensures termination.
- Skipping or including characters allows modification of the string without using extra loops.