# Remove “b” and “ac” from a given string

Given a string, eliminate all “b” and “ac” in the string, you have to replace them in-place, and you are only allowed to iterate over the string once. (Source Google Interview Question)

Examples:

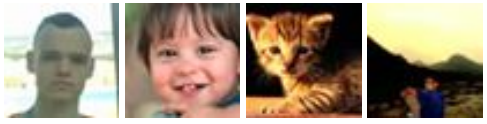
```
acbac    ==>  ""
aaac     ==>  aa
ababac   ==>   aa
bbbbd    ==>   d
```

The two conditions are:
**1.** Filtering of all ‘b’ and ‘ac’ should be in single pass
**2.** No extra space allowed.

The approach is to use two index variables i and j. We move forward in string using ‘i’ and add characters using index j except ‘b’ and ‘ac’. The trick here is how to track ‘a’ before ‘c’. An interesting approach is to use a two state machine. The state is maintained to TWO when previous character is ‘a’, otherwise state is ONE.
**1)** If state is ONE, then do NOT copy the current character to output if one of the following conditions is true
…**a)** Current character is ‘b’ (We need to remove ‘b’)
…**b)** Current character is ‘a’ (Next character may be ‘c’)
**2)** If state is TWO and current character is not ‘c’, we first need to make sure that we copy the previous character ‘a’. Then we check the current character, if current character is not ‘b’ and not ‘a’, then we copy it to output.

```
// A C++ program to remove "b" and 'ac' from input string
```





Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
#include <iostream>
using namespace std;
#define ONE 1
#define TWO 2

// The main function that removes occurences of "a" and "bc" in input
void stringFilter(char *str)
{
    // state is initially ONE (The previous character is not a)
    int state = ONE;

    // i and j are index variables, i is used to read next character o
    // string, j is used for indexes of output string (modified input
    int j = 0;

    // Process all characters of input string one by one
    for (int i = 0; str[i] != '\0'; i++)
    {
        /* If state is ONE, then do NOT copy the current character to
           one of the following conditions is true
           ...a) Current character is 'b' (We need to remove 'b')
           ...b) Current character is 'a' (Next character may be 'c')
        if (state == ONE && str[i] != 'a' && str[i] != 'b')
        {
            str[j] = str[i];
            j++;
        }

        // If state is TWO and current character is not 'c' (otherwise
        // we ignore both previous and current characters)
        if (state == TWO && str[i] != 'c')
        {
            // First copy the previous 'a'
            str[j] = 'a';
            j++;

            // Then copy the current character if it is not 'a' and 'b
            if (str[i] != 'a' && str[i] != 'b')
            {
                str[j] = str[i];
                j++;
            }
        }

        // Change state according to current character
        state = (str[i] == 'a')? TWO: ONE;
    }
```



## Popular Posts


- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

```
    // If last character was 'a', copy it to output
    if (state == TWO)
    {
        str[j] = 'a';
        j++;
    }

    // Set the string terminator
    str[j] = '\0';
}

/* Driver program to check above functions */
int main()
{
    char str1[] = "ad";
    stringFilter(str1);
    cout << str1 << endl;

    char str2[] = "acbac";
    stringFilter(str2);
    cout << str2 << endl;

    char str3[] = "aaac";
    stringFilter(str3);
    cout << str3 << endl;

    char str4[] = "react";
    stringFilter(str4);
    cout << str4 << endl;

    char str5[] = "aa";
    stringFilter(str5);
    cout << str5 << endl;

    char str6[] = "abababaac";
    stringFilter(str6);
    cout << str6 << endl;

    return 0;
}
```

Output:

```
ad
```

```
aa
ret
aa
aaa
```

**An extension of above problem where we don't want "ac" in output at all:**
The above code looks fine and seems to handle all cases, but what if input string is "aacacc", the above code produces output as "ac" which looks correct as it removes consecutive occurrences of 'a' and 'c'. What if the requirement is to not have an "ac" in output string at all. Can we modify the above program to produce output as empty string for input "aacacc" and produce output as "d" when input is "abcaaccd"? It turns out that it can also be done with given restrictions. The idea is simple. We need to add following lines inside for loop of the above program.

```
if (j>1 && str[j-2] == 'a' && str[j-1] =='c')
  j = j-2;
```

See this for different test cases of modified program.

This article is contributed by **Varun Jain**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above





## Recent Comments

affiszerv Your example has two 4s on row 3, that's why it...
Backtracking | Set 7 (Sudoku) · 13 minutes ago

**RVM** Can someone please elaborate this Qs from above...
Flipkart Interview | Set 6 · 33 minutes ago

**Vishal Gupta** I talked about as an Interviewer in general,...
Software Engineering Lab, Samsung Interview | Set 2 · 33 minutes ago

**@meya** Working solution for question 2 of 4f2f round....
Amazon Interview | Set 53 (For SDE-1) · 1 hour ago

sandeep void rearrange(struct node *head) {...
Given a linked list, reverse alternate nodes and append at the end · 2 hours ago

Neha I think that is what it should return as, in...
Find depth of the deepest odd level leaf node · 2 hours ago

## Related Tpoics:

- Printing Longest Common Subsequence
- Suffix Array | Set 2 (nLogn Algorithm)
- Rearrange a string so that all same characters become d distance away
- Recursively remove all adjacent duplicates
- Find the first non-repeating character from a stream of characters
- Dynamic Programming | Set 33 (Find if a string is interleaved of two other strings)
- Dynamic Programming | Set 29 (Longest Common Substring)
- Write your own atoi()



**Writing code in comment?** Please use **ideone.com** and share the link here.

**90 Comments** GeeksforGeeks

Sort by Newest



**Ankit Jain** • 13 days ago

we can do it with counter

```
void printWithoutBandAC(char str[])
{
int counter=0;
int i=0;
while(str[i])
{
if(str[i]=='b')
counter++;
else if(str[i]=='c')
```

{
str[i-counter]='c';
if(i-counter>0)
if(str[i-counter-1]=='a')
counter+=2;
}
else
{
str[i-counter]=str[i];
}
i++;

}
str[i-counter]=0;
printf("%s\n",str);
}



**nikola** • 21 days ago

This worked for me.. Working for all test cases... ("aacc",“abcaaccd”, etc ) Ple

void stringFilter(char *str)
{
int j=-1;

for(int i=0;str[i];i++)

{

if(str[i]=='b')

continue;

if( str[i] == 'c' && j >= 0 && str[j] == 'a' )

```
j--;

else
str[++j]=str[i];
}
str[++j]='\0';
}
```

Reply • Share ›

**nikola** • 21 days ago

This worked for me.. Working for all test cases... ("aacc",“abcaaccd”, etc ) Ple

```
void stringFilter(char *str)
{
int j=-1;

for(int i=0;str[i];i++)

{

if(str[i]=='b')

continue;

if(str[i]=='c'&&j>=0&&str[j]=='a')
j--;

else
str[++j]=str[i];
}
str[++j]='\0';
}
```

Reply • Share ›

**SunilVA** • 2 months ago

This looks to be much simpler solution:

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>

int main()
{

char strarray[][10] = {"acbac","aaac","ababac","bbbbd","ad","acbac","aaac","re

char *str = strarray[0];
int to = 0, from = 0;
int i=0, j = (sizeof(strarray)/sizeof(strarray[0]));

while( i < j)
{

str = strarray[i];
int to = 0, from = 0;
```

see more

Reply • Share ›

**dmonster** • 2 months ago

```
Java Implementation...
public class StripString

{

public static void main(String[] args)

{
```

```
String test1 = "acbac";

String test2 = "aaac";

String test3 = "ababac";

String test4 = "bbbbd";

System.out.println(strip(test1));
```

see more

• Reply • Share ›

**Vignesh A** • 3 months ago

My Simplified Version in C language

#include<stdio.h>

fun(char *s, int n)

{

int i;

for(i=0; i < n; i++)

{

if(s[i]== 'a')

continue;

else if (s[i]=='b' && s[i+1]== 'c')

{

see more

1 ^ | v • Reply • Share ›

**Sumit Khanna** • 3 months ago

http://ideone.com/efTQFW this approach filters for strings like babacc too

^ | v • Reply • Share ›

**shashi jey** • 4 months ago

```
#include <iostream>

#include<stdlib.h>

using namespace std;

void result(char *name,char out[],char str[]);

int main()

{

int i,j;

char name[200];

char out[200];

char str[200];

cout<<"enter the string"<<endl; cin="">>name;
```

see more

^ | v • Reply • Share ›

```
#include<stdio.h>
#include<string.h>
int main()
{ char str[50];
printf("enter a string\n");
gets(str);
int i, j;
if(str[0]=='a' && str[1]=='c')
j=0;
else
j=1;
for(i=1;i<=strlen(str);i++)
{
if((str[i]!='b')&& ((!(str[i]=='a' && str[i+1]=='c')&&!(str[i]=='c'&&str[i-1]=='a'))))
{
str[j]=str[i];
j++;
}
}
puts(str);
return 0;
}
```

2 ▲ | ▼ • Reply • Share ›

**Kaustubh** • 5 months ago

```
#include <stdio.h>
#include <string.h>

main()
{
char str[] = "ababaac" ;
```

```
for(i=0;str[i]!='\0';i++){
if(str[i] == 'b'){
str[i]='1';

}
else{
if(str[i] == 'a' && str[i+1] == 'c'){
str[i] = '1';
str[i+1] = '1';
}
}
```

see more

1 · Reply · Share ›

**groomnestle** · 5 months ago

Hack up below code which is more of my understanding of the original algorith

```
#include <iostream>

using namespace std;

int main()

{

char str[]="bababac";

int i, j =0;

int state = 1;

for(i=0;str[i]!='\0';i++)
```

{

// for non 'b', non 'a' element, add to string

see more

Reply • Share ›

**gopinath** • 6 months ago

write a program remove all '.' from inpu string.

Reply • Share ›

**groomnestle** gopinath • 5 months ago

To remove one character you don't need state machine at all.

```
int i, j = 0
for(i=0;str[i]!='\0';i++)
{
if(str[i]!='.')
{
str[j] = str[i];
j++;
}
}

str[j] = '\0';
```

1 Reply • Share ›

**Rahul** • 6 months ago

The program will not work if there is 'b' between a and c for example the outpu
the program prints "acde"

Reply • Share ›

**radhakrishna** • 6 months ago

Hi,

This is radhakrishna. Here is my solution..

```
public static void removebac(String ch) {

int j = 0;

char prev= ' ';

char[] s = ch.toCharArray();

for(int i =0 ; i< s.length; i++) {

if(s[i] == 'b' ) {

continue; // do nothing

} else if(s[i] == 'c' && prev == 'a') {

i--;
```

see more

• Reply • Share ›

**Brijesh Kumar Saini** • 7 months ago

above program will not work if str[0]='b'...i.e. if first character of string is 'b'...be conditions in loop will not be true..and it get passed...i.e. it will be like State=OI wrong

• Reply • Share ›

**Pushkar** ➦ Brijesh Kumar Saini • 14 hours ago

If str[0]='b' it means that we don't have to put it into our final string that' Suppose the string is like "bd" then after first iteration the loop goes into by d.

• Reply • Share ›

**archit** ➚ Brijesh Kumar Saini • 5 months ago

you are correct , even in case 'ac' it will not work

• Reply • Share ›

**Paparao Veeragandham** ➚ archit • 4 months ago

Hi

```
if (state == 1 && str[i] != 'b' && str[i] != 'a')
str[j++] = str[i];
//End for loop we need check below condition
if ( j == 0)
str[0] = '\0';
```

• Reply • Share ›

**Praneeth Reddy Bokka** • 7 months ago

```
package Strings;

public class RemovebacfromString {

public static void main (String [] args)

{

String s = "bbbbd";

boolean removedb = false;

for (int i =0; i<s.length();i++) {="" if="" (s.charat(i)="'b')" {="" s="s.substring(0,
continue;="" }="" if="" (i+1<s.length()="" &&="" s.charat(i)="'a'" &&="" s.chara
s="s.substring(0,i)+s.substring(i+2,s.length());" i="i-1;" }="" }="" system.out.pri
```

• Reply • Share ›

**Praneeth** Praneeth Reddy Bokka • 7 months ago

```
package Strings;

public class RemovebacfromString {

public static void main (String [] args)

{

String s = "bbbbd";

for (int i =0; i<s.length();i++) {="" if="" (s.charat(i)="'b')" {="" s="s.subs
i="i-1;" continue;="" }="" if="" (i+1<s.length()="" &&="" s.charat(i)="'a''
s="s.substring(0,i)+s.substring(i+2,s.length());" i="i-1;" }="" }="" system
```

Reply • Share ›

**devC#** • 8 months ago

```
public static void Remove(ref string str)

{

if(str == null || str.Length == 0)

return;

Char[] inputStr = str.ToCharArray();

int idxRead = 0;

int idxWrite = 0;

while (idxRead < inputStr.Length)

{

if (inputStr[idxRead] == 'b')
```

{

see more

Reply • Share ›

**devC++** • 8 months ago

void Remove(std::string& inputstr)

{

if(inputstr.length() == 0)

return;

char *ptrWrite = &inputstr[0];

char *ptrRead = &inputstr[0];

while(*ptrRead != '\0')

{

if(*ptrRead == 'b')

ptrRead++;

else if (*ptrRead == 'a' && *(ptrRead+1) == 'c')

see more

Reply • Share ›

**Pranshu Gupta** • 8 months ago

check this one out.....

#include<stdio.h>

void filter(char str[])

```
{ int i=0,j=0;
while(str[i]!='\0')
{
if(str[i]=='b')
i++;
else if(str[i]=='a' && str[i+1]=='c')
i=i+2;
else
{
str[j]=str[i];
j++;
i++;
}
}
str[j]='\0';
printf("%s",str);
}
int main()
{
char aa[]="ababaacacabaacb";
filter(aa);
return 0;
}
```



**Sanjay Agarwal** Pranshu Gupta • 6 months ago

Great !!!



**mrn** • 8 months ago

int index=-1;

int state=0;

for(int i=0;i<n;i++){ if(a[i]="='a')" {="" if(state="=1)" a[++index]="a[i];" state="1;" if(!(state="=1))" a[++index]="a[i];" state="0;" }="" else="" {="" if(state="=1)" a[+ )="" a[++index]="a[i];" }="" }="" a[++index]="\0" ;="">



**kju** • 8 months ago

void removechar(char *s)

{

int l=strlen(s);

int i,j=0;

for(i=0;i<l-1;) {="" if(s[i]="='a'" &&="" s[i+1]="='c')" {="" s[i]="1" ;="" s[i+1]="1" ;= s[i]="1" ;="" i+="1;" }="" else="" {="" printf("%c",s[i]);="" i++;="" }="" }="" }="">



**kju** • 8 months ago

```
void removechar(char *s)

{

    int l=strlen(s);

    int i,j=0;

    for(i=0;i<l-1;) {="" if(s[i]="='a'" &&="" s[i+1]="='c')" {="" s[i
```

**Adi** • 8 months ago

```
public class stringman {

public static String movement(int pos, String a){

int counter;

StringBuilder word = new StringBuilder(a);

char temp = a.charAt(pos);

for(counter = pos+1 ; counter< a.length(); counter++){

word.setCharAt(counter-1, a.charAt(counter));

}

word.setCharAt(counter-1, temp);

a = word.toString();

return a;
```

see more

Reply • Share ›

**Shiva Shankar Anumula** • 8 months ago

This problem can be solved with help of two pointers in more simplified way..

```
void RemoveAandBC(char *str)
{
int i=0;
char *q=str;
while(*q)
{
if(*q=='b')
```

```
q++;
else
{
if(*q=='c'&&(str[i-1]=='a'&&i>0))
{
q++;
i--;
}

else
```

see more

• Reply • Share ›

**Gamer12** • 8 months ago

```
int main(void)
{
char string[] = "react";
int i,j;
i = j = 0;
while( string[j] )
{
if( string[j] == 'b' )
j++;
else if( string[j] == 'a' && string[j+1] == 'c' )
j += 2;
else
{
string[i++] = string[j++];
}
}
string[i] = '\0';
```

```
printf("String: %s\n", string);
return 0;
}
```

Reply • Share ›

**ubiquitous** • 8 months ago

i felt this is exaggeration of technical word usage. It's not at all needed. We ca

Reply • Share ›

**yunhu** • 8 months ago

```
public static void removePattern(char[] s, out int len)

{

int i = 0; int j = 0;

while (j < s.Length)

{

if (s[j] == 'b')

++j;

else if (s[j] == 'c')

{

if (i != 0 && s[i-1] == 'a') --i;

else
```

see more

Reply • Share ›

**G2** • 8 months ago

Working JAVA solution :

static void replace(StringBuffer s){

int len = s.length();

for(int i=0; i < len;i++){

if(i < len - 1 && s.charAt(i) == 'a' && s.charAt(i+1) == 'c')

{

s.deleteCharAt(i);

s.deleteCharAt(i);

i += 1;

len -= 2;

continue;

see more

^ | v • Reply • Share ›

**Ranj** • 8 months ago

```
void func2(string str)

{int tail=0;

        for(int i=0;i<str.length();i++) {="" if(str[i]="'b')" continu
```

^ | v • Reply • Share ›

**Ranj** • 8 months ago

```
void func2(string str)

{int tail=0;

for(int i=0;i<str.length();i++) {="" if(str[i]="='b')" continue;="" if(str[i]="='a'" &&="
str[i+1]="='c')" {i++;="" continue;}="" else="" str[tail++]="str[i];" }="" str[tail]="\0'
cout<<"output="" is="" "<<str;="" }="" void="" main()="" {="" func2("bac1acac"
```

1 ▲ | ▼ • Reply • Share ›

**gawaskar** • 8 months ago

```
#include
#define LEN 100

void rem(char *str)
{
int i = 0, j = 0;

while(str[j]) {
if(str[j] == 'a') {
if(str[j+1] == 'c') {
j++;
} else {
str[i++] = str[j];
}
} else if(str[j] != 'b') {
str[i++] = str[j];
}
j++;
}
```

see more

▲ | ▼ • Reply • Share ›

In java, Strings are immutable, if I dont konw C much, how would i take this up



**dmonster** ➦ Nikhil • 2 months ago

Java Implementation...

```
public class StripString

{

public static void main(String[] args)

{

String test1 = "acbac";

String test2 = "aaac";

String test3 = "ababac";

String test4 = "bbbbd";

System.out.println(strip(test1));

System.out.println(strip(test2));
```

see more



**penhaunt** • 9 months ago

Why don't you this strategy which I got from ABHAY :)

```
#include<stdio.h>
void google(char *s)
{
```

```
    int i=0,j;
    for(j=0;s[j];j++)
    {
        if(s[j]!='b')
        {
            if(i>0&&s[j]=='c'&&s[i-1]=='a')
            i--;
            else
            s[i++]=s[j];
        }
    }
    s[i]='&#92&#48';
}
```

see more

• Reply • Share ›

**neelkand** • 9 months ago

[sourcecode language="C++"]
#include<iostream>
using namespace std;
void stringFilter(char *str){
int i,j=0,temp=0;
for(i=0;str[i]!='&#092&#048';i++){
if(str[i]=='a'){
if(temp==1)
str[j]='a';
temp=1;
}
else if(str[i]=='c' && temp==1){
j--;
temp=0;

```
continue;
}
else if(str[i]=='b'){
if(temp==1)
```

see more

Reply • Share ›

**animesh bhatia** • 9 months ago

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str[20],str1[20],str2[20];
char *s,*s1,*s2,*s3;
printf("\nEnter string:\n");
gets(str);
int i,j,k=0;
s=str;
s1=str1;
s3=str2;
while(*s)
{
if(*s=='b')
s=s+1;
```

see more

Reply • Share ›

**richa_431** • 9 months ago

//remove 'b' and 'ac' from a given string

```
#include<stdio.h>
#include<conio.h>

char * remove (char * s)
{

char * r;
int c=0;
while(s!='\0')
{
if(*s=='a'&&*(s+1)=='c')

{
s+=2;
continue;

}
```

see more

• Reply • Share ›

**AlgoCoder** • 10 months ago

```
void search_ac_b(char* str, int &i)
{
        if(str[i] == '&#92&#48')
                return;
        if(str[i] == 'b') search_ac_b(str, ++i);
        else if(str[i] == 'a' && str[i+1] == 'c')
        {
                i += 2;
```

```
                search_ac_b(str, i);
        }
        return;
}
void remove(char* str)
{
        int i = 0;
        int j = 0;
        if(str[i] == '&#92&#48')
```

see more

Reply • Share ›

**Nitin Panwar** • 10 months ago

```
#include<stdio.h>
#include<string.h>
void remove(char *);.
void set(char *, int, int);.
int main()
{
char aa1[]="acbac";.
char aa2[]="aaac";.
char aa3[]="ababac";.
char aa4[]="bbbbd";.
remove(aa1);.
remove(aa2);.
remove(aa3);.
remove(aa4);.
printf("%sn", aa1);.
printf("%sn", aa2);.
printf("%sn", aa3);.
printf("%sn", aa4);.
```

see more

Reply • Share ›

**shilendhari** • 10 months ago

a recursive solution

```
#include "stdio.h"
#include "stdlib.h"
void remove1(char *str)
{
        int i=0,flag=0;
        char temp;
        char* new;
        new=malloc(sizeof(char));
        char *new1 = new;
                while(*str!='&#92&#48')
                {
                        if(*str=='b')
                        {
                                str++;
                        }
```

see more

Reply • Share ›

**Bala** shilendhari • 9 months ago

Your code needs to increment flag even when you encounter 'b'. Else i 'abc'. But a neat recursive solution nevertheless !!

Reply • Share ›

**Swapnil Sunil Gondkar** • 10 months ago

read the question carefully it is given not to iterate over the string again !

Reply • Share ›

**Tapas Mahanta** • 10 months ago

#include<iostream>
#include<string>
using namespace std;.
int main()
{
string s="react";
while((s.find("ac"))!=s.npos)
s.erase(s.find("ac"), 2);
while((s.find("b"))!=s.npos)
s.erase(s.find("b"), 1);
cout<<s<<endl;
}
//tell me if any test cases don&#039t work on this one.

Reply • Share ›

**adwait** • 10 months ago

```
#include<stdio.h>
#include<string.h>
#define MAX 100
rmacb(char a[])
{
int pos,cur,n;
pos=cur=0;
n=strlen(a);
int i;
while(pos<n)
```

```
{
label1:
if(a[pos]=='b')
        pos++;
else
        if(a[pos]=='a'&&a[pos+1]=='c')
                pos=pos+2;
```

see more

• Reply • Share ›

Load more comments

Subscribe Add Disqus to your site