Remove characters from the first string which are present in the second string

Write an efficient C function that takes two strings as arguments and removes the characters from first string which are present in second string (mask string).

Algorithm: Let first input string be"test string" and the string which has characters to be removed from first string be "mask"
1: Initialize:
res_ind = 0 /* index to keep track of processing of each character in i/p string */
ip_ind = 0 /* index to keep track of processing of each character in the resultant string */

2: Construct count array from mask_str. Count array would be:
(We can use Boolean array here instead of int count array because we don't need count, we need to know only if character is present in mask string)
count['a'] = 1
count['k'] = 1
count['m'] = 1
count['s'] = 1

3: Process each character of the input string and if count of that character is 0 then only add the character to the resultant string.
str = "tet tringng" // 's' has been removed because 's' was present in mask_str but we we have got two extra characters "ng"
ip_ind = 11
res_ind = 9

4: Put a '\0' at the end of the string?

Google™ Custom Search

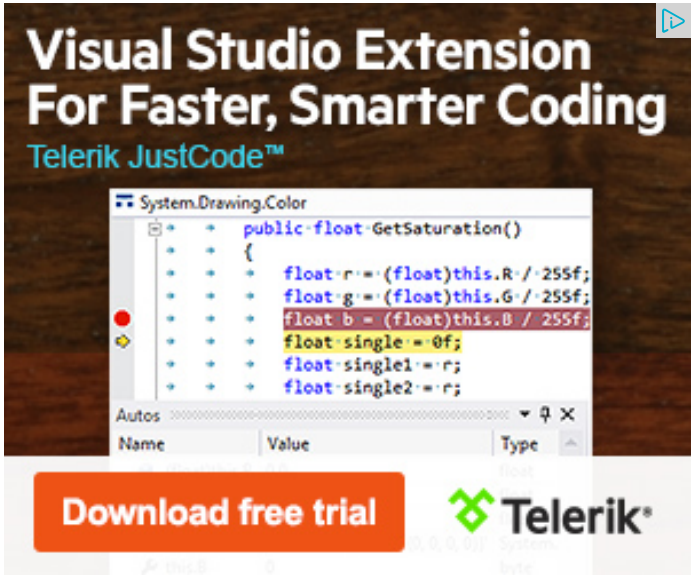
GeeksforGeeks

52,731 people like GeeksforGeeks.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Implementation:

```
#include <stdio.h>
#include <stdlib.h>
#define NO_OF_CHARS 256

/* Returns an array of size 256 containg count
   of characters in the passed char array */
int *getCharCountArray(char *str)
{
   int *count = (int *)calloc(sizeof(int), NO_OF_CHARS);
   int i;
   for (i = 0; *(str+i);  i++)
      count[*(str+i)]++;
   return count;
}

/* removeDirtyChars takes two string as arguments: First
string (str)  is the one from where function removes dirty
characters. Second  string is the string which contain all
dirty characters which need to be removed  from first string */
char *removeDirtyChars(char *str, char *mask_str)
{
  int *count  = getCharCountArray(mask_str);
  int ip_ind  = 0, res_ind = 0;
  char temp;
  while(*(str + ip_ind))
  {
    temp = *(str + ip_ind);
    if(count[temp] == 0)
    {
      *(str + res_ind) = *(str + ip_ind);
      res_ind++;
    }
    ip_ind++;
  }

  /* After above step string is ngring.
    Removing extra "iittg" after string*/
  *(str+res_ind) = '\0';

  return str;
}

/* Driver program to test getCharCountArray*/
int main()
```

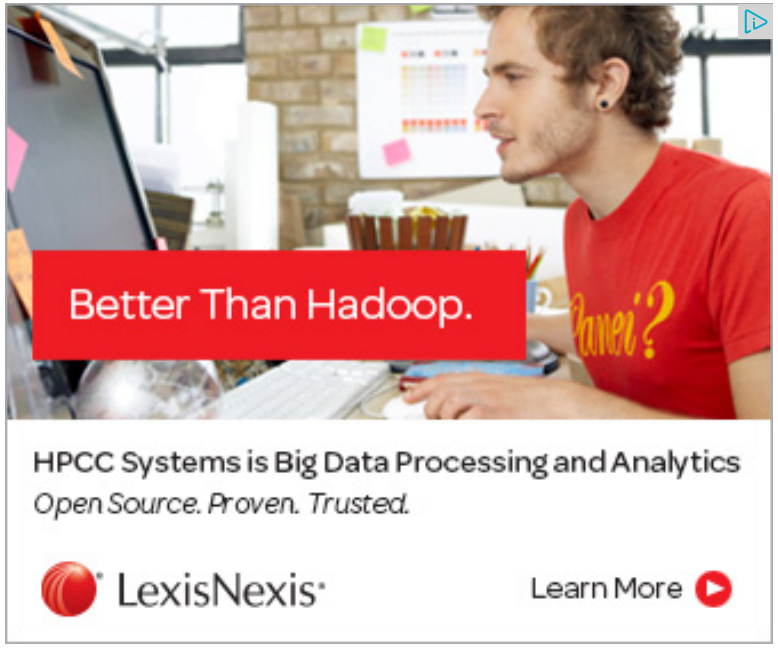
Popular Posts

- 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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
{
    char mask_str[]   = "mask";
    char str[]        = "geeksforgeeks";
    printf("%s", removeDirtyChars(str, mask_str));
    getchar();
    return 0;
}
```

Time Complexity: O(m+n) Where m is the length of mask string and n is the length of the input string.



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)
- Remove “b” and “ac” from a given string

ITT Tech - Official Site

itt-tech.edu

Associate, Bachelor Degree Programs Browse Programs Now & Learn More.

uStudio Video Platform

Pre-Launch App Testing

Agile Software Development

Watch Free PLC Tutorial

Html5 Tutorial

Just Strings

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

- Dynamic Programming | Set 29 (Longest Common Substring)

1 Tweet 0 0

695 Subscribe

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

24 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Abhi • 5 days ago

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

char* RemoveDupli(char* str1,char* str2)

{

char* str3=(char*)calloc(50,sizeof(char));

int i=0;

int j=0;

int x;

int flag=0;
```

Recent Comments

affiszerv Your example has two 4s on row 3, that's why it...

Backtracking | Set 7 (Sudoku) · 20 minutes ago

RVM Can someone please elaborate this Qs from above...

Flipkart Interview | Set 6 · 40 minutes ago

Vishal Gupta I talked about as an Interviewer in general,...

Software Engineering Lab, Samsung Interview | Set 2 · 40 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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

int len=strlen(str2);

see more

• Reply • Share ›

Satish Yadav • 9 months ago

A map could be used instead of a 256 size array...

```
 #include<iostream>
#include<map>
#include<string>

using namespace std;
typedef map<char, bool> charRecord;

string modifyString(string& test_str, string& mask_str)
{
        string final_str;
        charRecord record;
        //put mask string into map
        for(int i=0;i<mask_str.length();i++)
                record[mask_str.at(i)]=true;
        //check if the test string has any character form mask string
```

see more

1 • Reply • Share ›

abhishek08aug • a year ago

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

AdChoices
▶ Second String
▶ String Function
▶ Replace String
AdChoices
▶ C String
▶ String Java
▶ String Set
AdChoices
▶ Stream String
▶ String String
▶ New String

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
#define NO_OF_CHARS 256

char * mask_string(char * str, char * mask_str) {
  int * char_count=(int *)calloc(sizeof(int), NO_OF_CHARS);
  int current_copy_index=0;
  char * temp=str;
  while(*mask_str!='&#92&#48') {
    *(char_count+*mask_str)=*(char_count+*mask_str)+1;
    mask_str++;
  }

  while(*temp!='&#92&#48') {
    if(*(char_count+*temp)==0) {
      *(str+current_copy_index)=*temp;
      current_copy_index++;
```

see more

• Reply • Share ›

arpit tak • a year ago

[sourcecode language="java"]
public class RemoveFirstFromSecond {
public static void main(String[] args) {
String first = "arpit";
String second = "tp";
boolean bit[] = new boolean[256]; //boolean are defaulted to false

for(int i=0;i<second.length();i++){ .
bit[second.charAt(i)] = true;
}

char[] result = new char[first.length()];

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
for(int i=0,j=0;i<first.length();i++)
{
if(bit[first.charAt(i)]==false)
{
result[j] = first.charAt(i);
j++;
}

}

String resultstr= new String(result);
System.out.println("Result - " + resultstr);
}
}//:~
```

Reply • Share ›

Karthic • a year ago

Check the following link for java implementation :
http://myvedham.blogspot.in/20...

Reply • Share ›

Akshay Manathkar • a year ago

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

void rmovCharFromStr(char *str, char *mask_str).
{
int cntArr[256]={0};.

for(int I = 0; str[i]!= &#039&#039; i++).
cntArr[mask_str[i]]++;.
int I = 0;.
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

while(str[i]!=&#039&#039)
{.

if(cntArr[str[i]] == 0).

{.

cout<<str[i];.

---

see more

Reply • Share ›

poorvank • a year ago

Is my solution efficient?
#include<stdio.h>
#include<string.h>
int main()
{
int i,j,k;
char s1[100],s2[100];
printf("Enter the first string\n");
gets(s1);
printf("Enter the string you want the chars of to be removed :\n");
gets(s2);
for(i=0;i<strlen(s2);i++) {="" for(j="0;j<strlen(s1);j++)" {="" if(s2[i]="=s1[j])" {=
repetative="" characters="" with="" '#'="" }="" }="" }="" printf("final="" string="" i
{="" if(s1[i]!="#")="" printf("%c",s1[i]);="" }="" return="" 0;="" }="">

Reply • Share ›

datta • 2 years ago

/*Give the input strings as command line arguments and the program will rem
already present in string1*/

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
#include"stdio.h"
#include"stdlib.h"
#include"string.h"
#include"malloc.h"

int main(int argc,char **argv)
{
if(argc!= 3){
printf("Error:\nUsage\n./a.out \n");
exit(23);
}

int i;
char *str1,*str2,*p,*q,*z,*str3;
str1 = malloc(strlen(argv[1]));
str2 = malloc(strlen(argv[2]));
```

see more

• Reply • Share ›

karan • 3 years ago

R.Srinivasan & Ankit Mahendru:

Your approaches are same and O(n^2). The approaches given by @geeksforg

1 • Reply • Share ›

rajcools ➦ karan • 3 years ago

yes u r right but in geeksforgeeks solution there is space complexity of

• Reply • Share ›

radhakrishna ➦ rajcools • 3 years ago

can modify the solution to use hashing instead of count array to

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

string2 chracters in HashMap, and lookup for it during string1 tr

• Reply • Share ›

R.Srinivasan • 3 years ago

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

char * removeDirtyChars(char str[],char mask[])
{
     int i,j;
     for(i=0,j=0;str[i]!='';i++)
         if(strchr(mask,str[i])==NULL)
              str[j++]=str[i];
     str[j]='';
     return str;
}



int main() {
    char mask_str[]  = "mask";
    char str[]  = "geeksforgeeks";
    printf("%s", removeDirtyChars(str, mask_str));
    getchar();
    return 0;
}
```

• Reply • Share ›

prashant ➦ R.Srinivasan • 2 years ago

ur complexity is o(nm) where n and m are lengths of two strings since

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
/* Paste your code here (You may delete these lines if not wri
```

Reply • Share ›

wgpshashank • 3 years ago

@ankit xcellent work..

Reply • Share ›

Somnath • 4 years ago

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


void RemoveDup(char* str,char* mask)
{
        int mask_arr[256]={0};
        int mask_len=strlen(mask);
        int str_len=strlen(str);

        for(int i=0;i<mask_len;i++)
                mask_arr[mask[i]-'a']=1;

        int src=0;
        int dst=0;

        while(*(str+src))
```

see more

Reply • Share ›

Rider • 4 years ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

This is My solution:

```
char *AfterMask(char *first_string,char *mask_string){
        int chararray[256]={0};
        int i=0,k=0;
        char *final=(char *)malloc(sizeof(char)*strlen(first_string))
        for(i=0;i<strlen(first_string);i++)
                chararray[first_string[i]]++;
        for(i=0;i<strlen(mask_string);i++)
                chararray[mask_string[i]]=0;
        for(i=0;i<strlen(first_string);i++)
                if(chararray[first_string[i]]){
                        final[k++]=first_string[i];
                        chararray[first_string[i]--];
                }
        final[k]='';
        return final;
```

see more

Reply • Share ›

rv_10987 • 4 years ago

@ankit- I thinks that's the simplistic approach that is for each character in the string and update only if the character is not in the masked string. That will wo O(m.n) were m and n are the lengths of mask and input string respectively.

Reply • Share ›

Avi • 4 years ago

The solution provided by Ankit is O(n^3) (assuming m = n) in time because of loop is of O(n^2) time complexity if you observe it carefully :)

Reply • Share ›

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

WgpShashank ➦ Avi • 2 years ago

@Avi..SOln Provided by Ankit is not O(N^3) ..its Quadratic Only , checl
complexity ?
correct me if i am wrong ?

```
/* Paste your code here (You may delete these lines if not writ
```

^ | v • Reply • Share ›

geeksforgeeks • 4 years ago

@Ankit: Thanks for sharing your code. Could you please add few lines about a

^ | v • Reply • Share ›

jalajb2k7 ➦ geeksforgeeks • 2 years ago

his solution is of O(m*n). he is just searching for every traversed chara
Hash table instead

```
/* Paste your code here (You may delete these lines if not writ
```

^ | v • Reply • Share ›

Ankit Mahendru • 4 years ago

This is my solution :

```
#include<stdio.h>
int check(char s, char str[]);
int main()
{
  char str1[40], str2[5];
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
int i=0, j=-1, a;

printf("Enter the string you want to remove the chars from\n");
gets(str1);

printf("\n\n\Enter the string you want the chars of to be removed fr
gets(str2);

while(str1[i]!='&#92&#48')
{
```

see more

• Reply • Share ›

crazypro ➦ Ankit Mahendru • 2 years ago

@ankit:
your code is of (n*n) complexity code with the extra overhead of functic

• Reply • Share ›

Abhi ➦ Ankit Mahendru • 3 years ago

@ankit: very good code. Easily understandable

• Reply • Share ›

Subscribe Add Disqus to your site

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com