Print all the duplicates in the input string.

Write an efficient C program to print all the duplicates and their counts in the input string

Algorithm: Let input string be "geeksforgeeks"
1: Construct character count array from the input string.

count['e'] = 4
count['g'] = 2
count['k'] = 2
……

2: Print all the indexes from the constructed array which have value greater than 0.

Solution

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

/* Fills count array with frequency of characters */
void fillCharCounts(char *str, int *count)
{
   int i;
   for (i = 0; *(str+i);  i++)
      count[*(str+i)]++;
}

/* Print duplicates present in the passed string */
void printDups(char *str)
{
  // Create an array of size 256 and fill count of every character in
```

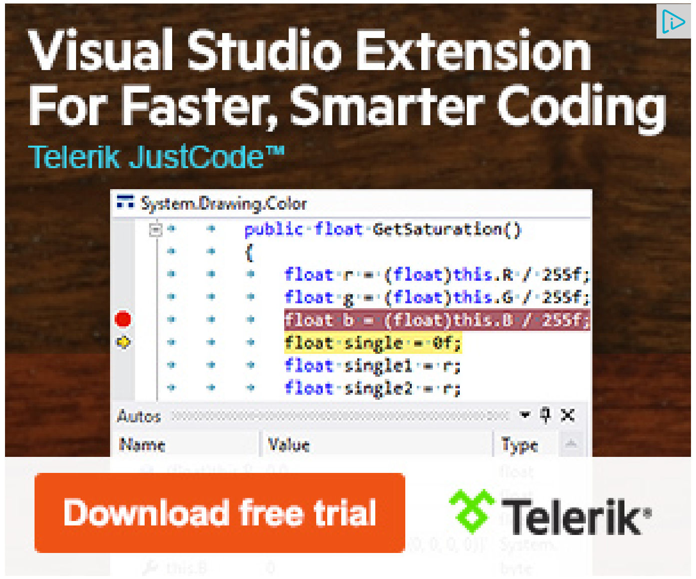
Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
  int *count = (int *)calloc(NO_OF_CHARS, sizeof(int));
  fillCharCounts(str, count);

  // Print characters having count more than 0
  int i;
  for (i = 0; i < NO_OF_CHARS; i++)
    if(count[i] > 1)
        printf("%c,  count = %d \n", i,  count[i]);

  free(count);
}

/* Driver program to test to pront printDups*/
int main()
{
    char str[] = "test string";
    printDups(str);
    getchar();
    return 0;
}
```

Output:

```
s,  count = 2
t,  count = 3
```

Time Complexity: O(n)



```
public float GetSaturation()
{
    float r = (float)this.R / 255f;
    float g = (float)this.G / 255f;
    float b = (float)this.B / 255f;
    float single = 0f;
    float single1 = r;
    float single2 = r;
```

Popular Posts

All permutations of a given string

Memory Layout of C Programs

Understanding “extern” keyword in C

HPCC Systems is Big Data Processing and Analytics
Open Source. Proven. Trusted.

- 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

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
- Dynamic Programming | Set 29 (Longest Common Substring)

4 Tweet 0 0

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

25 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Abhi • 8 days ago

```
#include<stdio.h>

#include<stdlib.h>

#define bool int

#define NO_OF_CHARS 256
```

```
void countDupli(char* str)

{

bool bin_bash[NO_OF_CHARS]={0};

int i;

int j;

char temp;

char* count=(char*)calloc(NO_OF_CHARS,sizeof(char));
```

see more

Reply · Share ›

Guest · 2 months ago

Hello @GeeksforGeeks-
This problem is similar to find duplicates in Integer Array.

Reply · Share ›

Sanjay Yadav · 7 months ago

```
#include<stdio.h>
int main()
{
int i=0,j=0;
int count[256]={0};
char str[50];
printf("Enter string\n");
```

695 Subscribe

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

```
gets(str);
while(*(str+i))
{
count[*(str+i)]++;
i++;
}
while(*(str+j))
{
if(count[*(str+j)]>1)
{
printf("%c,count=%d\n",*(str+j),count[*(str+j)]);
count[*(str+j)]=0;
}
j++;
}
return 0;
}
```

• Reply • Share ›

Utkarsh Agrawal • 7 months ago

Please fix the error in the statement:

2: Print all the indexes from the constructed array which have value greater thi

6 • Reply • Share ›

10bce0123 • 9 months ago

```
 #include<stdio.h>
#include<string.h>
int main()
{
        char *a="geeksforgeeks";
        int count[26]={0};
        int i;
        int len=strlen(a);
        for(i=0;i<len;i++)
        {
                count[a[i]-97]++;
        }
        for(i=0;i<26;i++)
        if(count[i]>1)
        printf("%c has occured %d times\n",i+97,count[i]);
        return 0;
}
```

Reply • Share ›

Sudheer Singampalli • 9 months ago

#include<conio.h>
#include <stdio.h>
int main()
{

char a[50];.

printf("enter a string n t");.

gets(a);.

char *p1,*p2;.

p1=a;.

while(*p1!=&#039&#039).

{.

char c=*p1;.

see more

Reply • Share ›

maheshgs • 10 months ago

2: Print all the indexes from the constructed array which have value greater thi

Should this be greater than 1 ?

Reply • Share ›

shruti • 11 months ago

Hi,

Instead of allocating memory for 256 integers(4 bytes) , can't we calloc for 256
instead of :int *count = (int *)calloc(NO_OF_CHARS, sizeof(int));
cant we use : char *count =(char *)calloc(NO_OF_CHARS, sizeof(char));
as of all the ascii codes can come in a single byte and we would be saving on

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

Reply • Share ›

Amaan → shruti • 10 months ago

I think that is correct. We can use sizeof(char) to save memory.

But to keep the semantics correct we use integer type, since we are s
exactly a character. So using char will save memory, but make code le
matter in such a small program obviously).

Reply • Share ›

Ronny → Amaan • 10 months ago

@Amaan

IMHO the use of int as hash is nowhere related to semantics. It
more than 127 occurrences of a character, then how are you g
in a char variable will lead to -128 and when you check for dupli
if(count[str[i]] > 1) it will result in FALSE despite the fact that it h

Reply • Share ›

Ronny shruti • 10 months ago

@shruti

The memory is allocated for keeping the count of occurances of that c
If we make it a char array, then the maximum value that can be stored
is used)
So it will overflow if occurance of a character is more than 255.

Reply • Share ›

Nikhil Gupta • 11 months ago

```
 /* Paste your code here (You may delete these lines if not writing co
char maxfreq(const char *str)
{
        int *arr;
        arr=(int*)malloc(256*sizeof(int));
        int i,index=0;
        for(i=0;i<256;i++)
        arr[i]=0;
        while(*str!='&#092&#048')
        {
                if(*str==' ' || *str=='\t' || *str=='\n');
                else
                index=++arr[*str]>arr[index]?(*str):index;
                str++;
        }
        return index;
}
```

^ | ⌄ • Reply • Share ›

Gupta • 11 months ago

Keep it Simple !!!! =D

```
#include
#include
#include
int main()
{
char str[]="Stupid programmer";
int count[255]={0},i,j=0,k=0,n;
char a[10];
printf("Duplicates in the given String %s are...\t",str);
for(i=0;*(str+i);i++)
{
if(count[*(str+i)]==0)
{
count[*(str+i)]++;
*(str+j)=*(str+i);
j++;
}
```

see more

^ | ⌄ • Reply • Share ›

abhishek08aug • a year ago

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

```
void print_duplicates(char * str) {
  int * char_count=(int *)calloc(sizeof(int), NO_OF_CHARS);
  int current_index=0;
  while(*str!='&#92&#48') {
    if(*(char_count+*str)==0) {
      *(char_count+*str)=*(char_count+*str)+1;
    } else {
      printf("Character %c at index %d is a duplicate\n", *str, currei
    }
    current_index++;
    str++;
  }
}
```

see more

^ | ⌄ • Reply • Share ›

neo • a year ago

for printing characters can we iterate only through string characters so that it t
is small

^ | ⌄ • Reply • Share ›

cyberWolf • a year ago

@GeeksForGeeks : Memory has not been freed using free() after using calloc

^ | ⌄ • Reply • Share ›

GeeksforGeeks ➜ cyberWolf • a year ago

Thanks for pointing this out. We have updated the code.

^ | ⌄ • Reply • Share ›

shrinivas • 2 years ago

Here use of calloc is wrong .
int *count = (int *)calloc(sizeof(int), NO_OF_CHARS);
it should be
int *count = (int *)calloc(NO_OF_CHARS,sizeof(int));
this is present throughout website.

Reply • Share ›

GeeksforGeeks ➦ shrinivas • 2 years ago

@shrinivas: Thanks for pointing this out. We will update the posts.

Reply • Share ›

don • 2 years ago

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

int main()
{
char a[]="aaaaahhhmmmdddzeeenfffgggkkb";
char b[12];
char c[10];
int i=0,j=0,k=0,d=0;
while(a[i])
{
b[k]=a[i];
d=1;
for(j=i;j<45;j++){
if(b[k]==a[j+1])
{
d++;
```

see more

Reply • Share ›

Sharadkumar • 2 years ago

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

using namespace std;

int main()

{

string s ="hhe";

int i=0,j=0;

for(i=0;i<s.length();++i)
j^=s.at(i);

cout<<char(j)<<endl;
```

see more

Reply • Share ›

Avinash • 2 years ago

@ap: you are correct. But you just have to ensure that you don't print duplicate

Reply • Share ›

ap • 3 years ago

In the code of printDups(char *str) instead of passing through the array, if u pa
efficient and the duplicate characters are printed in order of the string.

Please correct me if i am wrong in the efficiency case..

Thank you

• Reply • Share ›

wgpshashank • 3 years ago

i think it can be done in O(logn) if binary search is applied...

correct me if i m wrong

• Reply • Share ›

Sandeep ➜ wgpshashank • 3 years ago

@wgpshashank: Binary search can be applied if the input is sorted. He
the input is sorted, then we can remove duplicates in O(n) using a sim

• Reply • Share ›