Return maximum occurring character in the input string

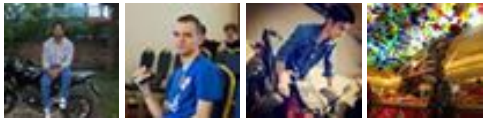
Write an efficient C function to return maximum occurring character in the input string e.g., if input string is “test string” then function should return ‘t’.

Algorithm:

```
Input string = “test”
1: Construct character count array from the input string.
  count['e'] = 1
  count['s'] = 1
  count['t'] = 2

2: Return the index of maximum value in count array (returns ‘t’).
```

Implementation:

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

int *getCharCountArray(char *);
char getIndexOfMax(int *, int);

/* Returns the maximum occurring character in
 the input string */
char getMaxOccuringChar(char *str)
{
  int *count = getCharCountArray(str);
```

```
  return getIndexOfMax(count, NO_OF_CHARS);
}

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

  for (i = 0; *(str+i);  i++)
    count[*(str+i)]++;

  return count;
}

char getIndexOfMax(int ar[], int ar_size)
{
  int i;
  int max_index = 0;

  for(i = 1; i < ar_size; i++)
    if(ar[i] > ar[max_index])
      max_index = i;

  /* free memory allocated to count */
  free(ar);
  ar = NULL;

  return max_index;
}

int main()
{
  char str[] = "sample string";
  printf("%c", getMaxOccuringChar(str));

  getchar();
  return 0;
}
```

Time Complexity: O(n)

Notes:
If more than one character have the same and maximum count then function returns only the first character in alphabetical order. For example if input string is “test sample” then function will

Popular Posts

return only ‘e’.

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)

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

62 Comments GeeksforGeeks

Sort by Newest

Sort by Newest

Join the discussion…

deepak • 2 months ago

yeah....i cannot able to understand how one can use character as index of arra

```
char *text = "this";

int i=0;

int *count = (int *)calloc(256, sizeof(int));

for (i = 0; i<21; i++)

count[*(text+i)]++;

for (i = 0; i<21; i++)

printf("\n %c = %d",*(text+i), count[*(text+i)]);

output : t = 1

h = 1

i = 1
```

see more

^ | v • Reply • Share ›

sachi059 • 2 months ago

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

Can we apply this logic after sorting the input string ...
1> Sort the string
2> Count which is repeated most and save which makes greater counts and t

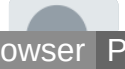
```
int cmpstringp( const void *p1, const void *p2)

{

int f = *((char*)p1);

int s = *((char*)p2);

if (f > s) return 1;

if (f < s) return -1;

return 0;

}

char max_repeated_char_logic(char *str)
```

see more

1 ^ | v • Reply • Share ›

Guest • 2 months ago
@Geek
^ | v • Reply • Share ›

Guest • 2 months ago
@Geeksforgeeks I think there should be a correction required.
Check it for "zz sample string"
^ | v • Reply • Share ›



rDx_1407 • 3 months ago

Here is my solution to this problem

package Strings;

public class max_occur_char {

public static void main(String[] arg)

{

String str="abcaba";

int count=0;

int big=0;

int index=0;

for(int i=0;i<str.length();i++) {="" for(int="" j="i+1;j<str.length();j++)" {="" if(str
count++;="" }="" }="" if(big<count)="" {="" big="count;" index="i;" }="" }="" syste
occuring="" character="" is:"+str.charat(index));="" }="" }="">

Marsha Donna • 3 months ago

because the max occuring character can only be 1 of the characters present i
function can also be written as

int getIndexofmax(int count[],char str[],int len) //where len is the of input string
{
int i,max=0,index=0;

for(i=0;i<len;i++) {="" if(count[str[i]]="">max)
{
max=count[str[i]];
index=i;

index=i;
}
}
return index;
}
pls correct me if anything is wrong

• Reply • Share ›

Mathiazhagan • 4 months ago

Can I solve this like this?

```
#include<iostream.h>
#include<conio.h>
int main()
{
char arr[100];
int count[256],max=0,position=0;

for(int i=0;i<256;++i)
count[i]=0;

cin>>arr;

for(int i=0;i<strlen(arr);++i) {="" count[arr[i]]++;="" }="" for(int="" i="0;i&lt;256;+
max="count[i];" position="i;" }="" }="" cout<<(char)(position);="" getch();="" ret
```

2 • Reply • Share ›

aayush • 5 months ago

```
#include<iostream>

#include<string>

#include<cstdio>
```

```
using namespace std;

main()

{

string s;

cin>>s;

int len;

len=s.length();

int a[256]={0};
```

see more

• Reply • Share ›

Guest • 5 months ago

lets have a simpler one and easy to understand -_-

make sure you use getline or gets if you are using space( )

here's the code

```
#include <iostream>

#include<string.h>

using namespace std;

char maxs(char*,int);

int main()

{
```

{

char s[20];

cin>>s;

see more

1 ^ | v • Reply • Share ›

Marsha Donna • 7 months ago

here is d implementation in c++
#include<iostream.h>
#include<string.h>

void maxocrcharinstr(char str[])
{

int count[256]={0},i;
for(i=0;i<strlen(str);i++) {="" count[str[i]]++;="" }="" int="" max="count[0];" char
if(count[i]="">max)
{max=count[i];
c=(char)i;
}
}

cout<<"max ocr chr is "<<c; }="" int="" main()="" {="" char="" str[]="sample st
getchar();="" return="" 0;="" }="" pls="" corrct="" me="" if="" any="" mistakes=

^ | v • Reply • Share ›

kuldeepshandilya • 9 months ago

Since we only need to know the maximum occurring character, we can maint
comparing nodes would be two step -
1) no of occurrence of character
2) If multiple characters have same no of occurrence, use their alphabetical ra

Each node in heap will contain two data - character and its occurrence.

Space complexity wise It is better than count array, bcoz count array will end u
characters which might not be even present in input string. Max heap will have

Rohit • 9 months ago

I think this prints the first most repeated character without unnecessarily loopir
anything is wrong.

```
include
#include
#define NO_OF_CHARS 256

int getCharCountArray(char *str) {
int *count = (int *) calloc(NO_OF_CHARS + 1, sizeof(int));
int i;
int maxIndex = NO_OF_CHARS; // a non- character in ascii set
count[maxIndex] = -1;
for (i=0; *(str+i); i++){
count[*(str + i)]++;
if((count[maxIndex] < count[*(str+i)]))
maxIndex = *(str+i);
}
free(count);
count = NULL;
return maxIndex;
}
int main()
{
char str[] = "test sample";
printf("%c", getCharCountArray(str));
```

return 0;
}

 • Reply • Share ›

NISHANT GUPTA • 9 months ago

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

why is the running time of my code O(n^2) while for yours its O(n)????

#include
#include
int main()
{
char a[50],b[50];
int i=0,ctr[50],x,max,k,len=0,temp,temp2;
scanf("%s",a);
for(k=0;k<strlen(a);k++)
ctr[k]=0;
while(a[i]!=&#039&#039)
{
x=0;
for(k=0;k<len;k++)
{

see more

 • Reply • Share ›

Vivek Venkatesh • 10 months ago

You can also use a HashMap to achieve the same thing. You don't need an ar
of the map will be equal to the number of unique characters in the string.

Time Complexity = O(n) + O(m) ~= O(n) [Since n > m]

where m is the number of entries (or number of distinct characters in the strin
traverse the HashMap and find the maximum value.

```
char findMaximumOccurenceCharacter() {
                HashMap<Character, Integer> countHash = new HashMap<Cl
                for(int i=0;i<input.length();i++) {
                                char currentChar = input.charAt(i);
                                if(countHash.get(currentChar) == null
                                        countHash.put(currentChar, 1)
                                }
                                else{
                                        countHash.put(currentChar, cou
                                }
                }
```

see more

magnet • 10 months ago

running time of for loop can be reduced to length of string if string length is less
functions

```
 int getCharCountArray(char *str)
{
  int *count = (int *)calloc(NO_OF_CHARS, sizeof(int));
  int i;

  for (i = 0; *(str+i);  i++)
    count[*(str+i)]++;
  int max_index = 0,max=0;

  for(i = 0; *(str+i); i++)
    if(count[*(str+i)] > max)
```

```
    if(count[*(str+i)] > max)
     max= count[*(str+i)],max_index = i;

  /* free memory allocated to count */
  free(count);
  count = NULL;

  return *(str+max_index);

}
```

• Reply • Share ›

kartheek • 11 months ago

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

void add_char_to_list(char ch);
void find_biggest();

struct char_count
{
        char charecter;
        int count;
        char_count *next;
}*head = NULL;

int main()
{
        char input[] = "jhkjshkjc uereiwjrvijrjv u uurijijf iewuhjauqv
        int len = 0,i;
```

see more

shruti • 11 months ago

Hi,

Can someone explain that in the code snippet given below:
char getIndexOfMax(int ar[], int ar_size)
{
int i;
int max_index = 0;

for(i = 1; i ar[max_index])
max_index = i;

/* free memory allocated to count */
free(ar);
ar = NULL;

return max_index;
}

since we are freeing the dynamically allocated pointed to by ar(which is also p
how are we able to access the memory present at max_index in the array cou
been freed. Shouldnot free be done after accessing the memory at max_index
my understanding is not correct.

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


Sudarshan Singh shruti • 11 months ago

count has already been used before ar has been freed and after gettin

abhishek08aug • a year ago

As they say "Sweet and Simple": :P

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

char get_maximum_occurring_character(char * str) {
  int * char_count=(int *)calloc(sizeof(int), NO_OF_CHARS);
  int max_count=0;
  char max_character='&#92&#48';
  while(*str!='&#92&#48') {
    *(char_count+*str)=*(char_count+*str)+1;
    if(*(char_count+*str)>max_count) {
      max_count=*(char_count+*str);
      max_character=*str;
    }
    str++;
  }
```

see more

Divy • a year ago

#include <stdio.h>
#include <conio.h>
void findmax(char *str1);
void findmax(char *str1)
{
int temp_char[256]={0};

printf("\n%s",str1);
for(int i=0;str1[i];i++)
{
temp_char[*(str1+i)]= temp_char[*(str1+i)]+1;
}
int max_index,j;
int max = temp_char[0];
for(j=1;j<256;j++)
{
if(max<temp_char[j]) {="" max="temp_char[j];" max_index="j;" }="" }="" printf("
times="" in="" %s",max_index,max,str1);="" }="" int="" main()="" {="" char="" s
printf("%s",str);="" findmax(str);="" getch();="" return="" 0;="" }="">

• Reply • Share ›

pavi • a year ago

A small optimization:
Instead of having 2 functions:
int *getCharCountArray(char *);
char getIndexOfMax(int *, int);

we can push the logic of getIndexOfMax() inside the function: int *getCharCou
and changing its signature to int *getCharCountArray(char *str, int *maxIndex)
The function still returns the count array,
but now takes a pointer type to store the index of highest frequency char with-i

```
int *getCharCountArray(char *str, int *maxIndex) {
        int *count = (int *) calloc(NO_OF_CHARS + 1, sizeof(int));
        int i;
        *maxIndex = NO_OF_CHARS; // a non- character in ascii set
        count[*maxIndex] = -1;
        for (i=0; *(str+i); i++){
                count[*(str + i)]++;
```

```
            if((count[*maxIndex] < count[*(str+i)])
```

see more

• Reply • Share ›

pavi ➚ pavi • a year ago

Follow up.
since now both the logics of:

```
 int *getCharCountArray(char *);
char getIndexOfMax(int *, int);
```

have been clubbed together within one function:

```
getCharCountArray();
```

so we no more need the count array as return. Hence instead we can
So now the signature becomes:

```
int getCharCountArray(char *str);
```

• Reply • Share ›

nikinjain • a year ago

It's Important that we make a call to calloc here, because malloc doesn't Initial
to zero, while a call to calloc does that.

• Reply • Share ›

sush • a year ago

wrong output for "test sample".

• Reply • Share ›

GeeksforGeeks ➚ sush • a year ago

@sush: The program produces 'e' as output. There seems to be two c
The program prints one of them.

^ | v • Reply • Share ›

sush ➚ GeeksforGeeks • a year ago

but problem says it will print the first character. so it should prin

^ | v • Reply • Share ›

GeeksforGeeks ➚ sush • a year ago

@sush: Thanks for pointing this out. The program prints
order. We have updated the note. The solution can be e
character in string as well.

^ | v • Reply • Share ›

vijay • 2 years ago

what if two such characters are possible...that test case is left in this algo...for
to sort the array???

^ | v • Reply • Share ›

Mounika • 2 years ago

Why is the count array size 256? Any logic behind it. Isn't it enough if we just a

^ | v • Reply • Share ›

Sandeep ➚ Mounika • 2 years ago

Because, there are 0-255 ascii codes

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

^ | v • Reply • Share ›

Aashish ➚ Sandeep • 2 years ago

We can reduce the space by using bit-vector.

^ | v • Reply • Share ›

anonymous ➚ Aashish • a year ago

but how we cant have any other value except 0,1 so how can you take record of its occurences which will be more than 1

^ | v • Reply • Share ›

aaa • 2 years ago

```
import java.io.*;
import java.util.Scanner;

public class SearchCharater
{
public static void main(String args[])
{
String inputString,str;
int current,count=0;
char key;

Scanner keyborad=new Scanner(System.in);

System.out.println("Enter first String: ");
inputString=keyboard.nextLine();

System.out.println("Enter character :");
str=keyborad.nextLine();
key=str.charAr(0);
```

see more

^ | v • Reply • Share ›

Prashanth • 2 years ago

```
import java.util.*;

public class MaxOccurChar {
        public static List<Character> MaxChar(String inp){
                //Convert input String to char array
                char a[] = inp.toCharArray();
                //intialize an integer array of size 256
                int intA[]= new int[256];
                for(int i=0;i<intA.length;i++)
                        intA[i]=0;
                //scan the char array and count number of occurances
                for(char c : a)
                {
                        int x = c;
                        intA[x]++;
                }

                int max = 0;
```

see more

• Reply • Share ›

Prashanth ↗ Prashanth • 2 years ago

This code handles duplicates smoothly.....please comment if I am wro

• Reply • Share ›

Arya • 2 years ago

#include
#include
#include

```
#define NO_OF_CHARS 256
int main()
{
char str[100],ch[100];
int i,max=0;
gets(str);
int len=strlen(str);
int *count = (int *)calloc(NO_OF_CHARS, sizeof(int));
for(i=0;i<len;i++)
{
count[str[i]]++;
}

for(i = 0; i max)
max=i;

printf("%c\n",str[max]);
return 0;

}
```

^ | v • Reply • Share ›

sachin0382 • 3 years ago

your program is not working if we take input as
"sample STRING"

^ | v • Reply • Share ›

kartik ➚ sachin0382 • 3 years ago

The program actually prints space character (' ')

In your example, all characters of the input string are different. In cases have smae number of occurrences, the program prints the character v space in your example. See the output for "sampleSTRING" is 'G' whic

elham • 3 years ago

hi ,i want ur help
for a text ,this algorithm "the
occurrence of first letter in each double letter word is counted in each group ar
(MOFL) is
formed" in matlab ,i dont know how can write this program,plz help me.thanks

Prashant • 3 years ago

This uses a HashMap to store the characters in a String. It can be more space
array (size 256) is actually not fully utilized. Ofcourse, there is additional overh
an array of Entry objects but if the number of distinct characters is very few, th

```
[sourcecode language="java"]
public static char getMaxOccuringChar(String input) {
char[] inputArray = input.toCharArray();
HashMap charMap = new HashMap();
for(int i=0; i<inputArray.length; i++) {
if(charMap.get(inputArray[i]) == null)
{
charMap.put(inputArray[i], 1);
}
else {
int value = (Integer)charMap.get(inputArray[i]);
charMap.put(inputArray[i], value + 1);
}
}
```

Abhishek ➚ Prashant • a year ago

Hi,

I like your solution. We can optimize it further, by adding two variables.

When we change the value for a particular character, if its greater than that value, and MaxSumCharacter the Character for which the conditio

this way, we dont need to iterate through all the sum values again.

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

  • Reply • Share ›

sutendra mirajkar • 3 years ago

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

int main()
{

    char str[30];
    int count[30]={0};

    printf("ENTER THE STRING U WANT\n");
    scanf("%s",str);

    int i=0;

    while(str[i])
    {
```

```
    int n=str[i]-97;
```

see more

Reply • Share ›

sachin0382 sutendra mirajkar • 3 years ago

program is o.k. but will not work if string is in upper case so use this
n=(str[i]>=97)?str[i]-97:str[i]-65;
instead of
n=str[i]-97;

Reply • Share ›

yeskay • 3 years ago

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

#define ALPHA 26
int nchars[26];

int main(){
        char str[]="geeksforgeeks";
        int len = strlen(str);
        int i=0,ptr=0,max=0;
        for(i=0;i<len;i++)
                nchars[str[i] - 'a']++;

        max = nchars[0];

        for(i=1;i<ALPHA;i++)
                if(nchars[i] > max)
                        max = nchars[ptr=i];
```

```
        printf("The character %c is occured %d times \n",'a'+ptr,max)
getch();
return 0;
}
```

Reply • Share ›

RR • 3 years ago

The function getIndexOfMax() is supposed to return a char value whereas it is with highest freqency.

Could you just take a look ?

Reply • Share ›

kartik ➚ RR • 3 years ago

The index of count array won't be more than 255 and characters thems returning index as character should not be a problem in this program.

Reply • Share ›

Venki • 4 years ago

:) The stdlib.h and expression terminator (semicolon) are missing.

Reply • Share ›

GeeksforGeeks ➚ Venki • 4 years ago

OK, we have fixed them :)

Reply • Share ›

Gamer ➚ GeeksforGeeks • 3 years ago

The syntax of calloc is in correct. it is size first and then the dat

int *count = (int *)calloc(sizeof(int), NO_OF_CHARS);

Correct form

int *count = (int *)calloc(NO_OF_CHARS, sizeof(int));

• Reply • Share ›

Sandeep Gamer • 3 years ago

Thanks for pointing this out. We have fixed it.

• Reply • Share ›

Load more comments

Subscribe Add Disqus to your site