LEARN STRINGS, ARRAYS & FUNCTIONS IN C

Kaustav Ghosh Dastidar

GIZMOFACTS
www.gizmofacts.com

Learn Strings, Arrays & Functions In C

Introduction

This book primarily explains the essential parameters that are most used in the C programming language. It is an extremely user-friendly book for anyone just starting out in the world of coding!

C language is considered the mother language of all modern programming languages because most compilers, JVMs (Java Virtual Machines), Kernels, etc. are written in C language, and most programming languages, such as C++, Java, C#, PHP, JavaScript, Python, etc., follow C syntax.

With plenty of examples, illustrations, captions, code, and so on, this book will cover all of the fundamentals of the C language and programming in general. Almost every chapter will conclude with a related problem that you have to complete and, most importantly, read the explanation for each answer. You should practice more to grasp the programming concepts.

In this book, you can learn C programs on various topics such as strings, series, arrays, functions, loops, geometrical figures, mathematical calculation, sorting & searching algorithms, and many more. We hope to provide you with a good packaged solution under one roof. For all possible

C programming questions that you may have encountered in interviews or other assignments or projects Have fun coding!!

What you'll discover:

- Learn the fundamentals of the C programming language from the scratch.
- Learn the fundamentals of C programming, such as datatypes, variables, statements, loops, functions, strings, arrays, and pointers.
- You should be able to understand the concepts and write a C programme on your own by the end of this book.
- Learn one of the most widely used and popular programming languages in the software industry.

This book is intended for programmers, students taking other computer courses, and others who want to understand the workings of the most popular computer language, C.

About The Author

The man behind **Gizmofacts, Kaustav Ghosh Dastidar** holds a Master Degree in Computer Science. He is usually turned on (metaphorically) by technology and gadgets. Born and raised in a quintessential middle class family he has been well aware of the ignorance the mass harbours about technology. Through Gizmofacts he wants to reach out to all those people, who he believes just need a little push to get into this unique and amazing world of science and software.

Moreover, Kaustav is well aware that nurturing an interest in gadgets doesn't come cheap. Hence he wants to also be an enabler who would provide all the 'need to know' financial details of different gadgets so that people can live their dreams remaining in their budget.

To know more about tips and tricks of softwares, gizmos and mobile apps, follow him in ***Twitter***, ***Facebook*** and Google+.

You may also subscribe to Gizmofacts in ***Youtube*** for getting more information about software tips & tricks.

Disclaimer

This eBook may not be copied or distributed without permission in any way. This publication's content is offered solely for informational reasons. The usage or misuse of this eBook, as well as any financial damage incurred by individuals or property as a direct or indirect result of using this eBook, are not the author's responsibility.

We are unable to guarantee your success or outcomes in the future due to some unforeseeable risks associated with doing business online. You accept that the author is not responsible for any success or failure of your business that is related in any way to the download and use of our information and that the use of our information should be based on your due diligence.

Without the author's prior written consent, no portion of this eBook may be copied or otherwise distributed in any way, including electronically, mechanically, by photocopy, recording, or any other method

Table of Content

C Program-
Convert String To Lowercase

Code description

This "**C program**", will ask a user to enter a string (which can be in fully uppercase or partial uppercase characters) and then the program would convert it into a complete "***lower case string***". The logic behind the program is: All the upper case characters (A-Z) have ASCII value ranging from **65** to **90** and their corresponding lower case characters (a-z) have ASCII value ***32*** greater than them. For example '**A**' has a ASCII value **65** and '**a**' has a ASCII value **97** ***(65+32)***. This logic applies for other characters as well.

Also note, we have used "**string.h**" header library, since we have used "***strlen()***" library function (defined in the "**string.h" library**) to get the input string length in integer by counting number of characters.

Source code

```
//  Created by Kaustav Ghosh Dastidar


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


int main(){
//This main() function has 'integer' return type but you can also use'void' for no return
   char str[30];
   /* This array can hold a string of upto 30
    * chars, if you are going to enter larger string
    * then increase the array size accordingly
    */
   int i;
   printf("Enter the string: ");
   scanf("%s",str);
```

```
 for(i=0;i<=strlen(str);i++){
       if(str[i]>=65&&str[i]<=90)
           str[i]=str[i]+32;
    }
    printf("\nLower Case String is: %s",str);
    return 0;
}
```

Output

Enter the string: GIZMOFACTS

Lower Case String is: gizmofacts

C Program-
Convert String To Uppercase

Code Description

This “C program“, will ask a user to enter a string (which can be in fully lowercase or partial lowercase characters) and then the program would convert it into a complete “upper case string”. The logic behind the program is: All the lower case characters (A-Z) have ASCII value ranging from 97 to 122 and their corresponding upper case characters (a-z) have ASCII value 32 less than them. For example ‘a‘ has a ASCII value 97 and ‘A‘ has a ASCII value 65 (97-32). This logic applies for other characters as well.

Also note, we have used “string.h” header library, since we have used “strlen()”library function (defined in the “string.h” library) to get the input string length in integer by counting number of characters.

Source code

```
  //  Created by Kaustav Ghosh Dastidar.

#include<stdio.h>

#include<string.h>

int main(){

    char str[30];

    int i;

    printf(“Enter the string:”);

    scanf(“%s”,str);
```

```
for(i=0;i<=strlen(str);i++){
        if(str[i]>=97&&str[i]<=122)
            str[i]=str[i]-32;
    }
    printf("\nUpper Case String is: %s",str);
    return 0;
}
```

Output

```
Enter the string:gizmofacts
Upper Case String is: GIZMOFACTS
```

C-Program-
String Comparison

Code description

Here in the below code we are just taking two string as input. Then we are running a "for-loop" to the end of string and comparing each character of first string with the second string. The logic which has been implemented is by checking if the characters of the "***first string***" is greater than the "***second string***" then we are setting the "***flag value as 1***".

If both the character length count of the two strings are **identical** then we are setting the "**flag value as 0**". Finally, if the character length count of the second string is greater than the first string then we are setting the "**flag value as 2**".

Then based on the value of the "flag" **we are printing the outputs.**

Source code

```
// Created By Kaustav Ghosh Dastidar

#include<stdio.h>

void main()
{
        char temp,a[50],b[50];
        int i,j,k,flag=0;
      printf("Enter the first string: ");
      scanf("%s",a);
      printf("Enter the second string: ");
      scanf("%s",b);
                for(i=0;a[i]!='


// Created By Kaustav Ghosh Dastidar
#include<stdio.h>
void main()
{
char temp,a[50],b[50];
```

```
int i,j,k,flag=0;
printf("Enter the first string: ");
scanf("%s",a);
printf("Enter the second string: ");
scanf("%s",b);
for(i=0;a[i]!='\0'||b[i]!='\0';i++)
{
if(a[i]>b[i])
{
flag=1;
break;
}
}
if(flag==0)
printf("Two strings are equal....");
if(flag==1)
printf("First string is greater than second string....");
if(flag==2)
printf("Second string is greater than first string....");
}
'||b[i]!='


// Created By Kaustav Ghosh Dastidar
#include<stdio.h>
void main()
{
char temp,a[50],b[50];
int i,j,k,flag=0;
printf("Enter the first string: ");
scanf("%s",a);
printf("Enter the second string: ");
scanf("%s",b);
for(i=0;a[i]!='\0'||b[i]!='\0';i++)
{
if(a[i]>b[i])
{
```

```
flag=1;
break;
}
}
if(flag==0)
printf("Two strings are equal....");
if(flag==1)
printf("First string is greater than second string....");
if(flag==2)
printf("Second string is greater than first string....");
}
';i++)
                {
                 if(a[i]>b[i])
                  {
                    flag=1;
                        break;
                  }
                }
                if(flag==0)
                   printf("Two strings are equal....");
                if(flag==1)
                   printf("First string is greater than second string....");
                if(flag==2)
                   printf("Second string is greater than first string....");
}
```

C Program-Concatenate Two Strings Without Using strcat()

Code description

This is a simple "**C Program**" which is actually calculating the length of the "**first string**" and storing it in integer variable by counting the number of characters in "**first string**" using the "**first for-loop**".Then we are running a second "**for-loop**" to concatenate each characters of the "**second string**" at the end of "**first string**".

Source code

```
// Created By Kaustav Ghosh Dastidar

#include <stdio.h>
void main()
{
   char str1[50], str2[50], i, j;

   printf("Enter first string: ");
   scanf("%s", str1);

   printf("Enter second string: ");
   scanf("%s", str2);

   // calculate the length of string str1
   // and store it in integer variable
   // you can also use strlen() library function to calculate the string length
   for(i = 0; str1[i] != '
```

```
// Created By Kaustav Ghosh Dastidar
#include <stdio.h>
void main()
{
char str1[50], str2[50], i, j;
printf("Enter first string: ");
scanf("%s", str1);
printf("Enter second string: ");
scanf("%s", str2);
// calculate the length of string str1
// and store it in integer variable
// you can also use strlen() library function to calculate the string length
for(i = 0; str1[i] != '\0'; i++);
for(j = 0; str2[j] != '\0'; j++, i++)
{
str1[i] = str2[j];
}
str1[i] = '\0';
// '\0' represents the end of string
printf("String After concatenation: %s", str1);
}
'; i++);

    for(j = 0; str2[j] != '
```

```
// Created By Kaustav Ghosh Dastidar
#include <stdio.h>
void main()
{
char str1[50], str2[50], i, j;
printf("Enter first string: ");
scanf("%s", str1);
printf("Enter second string: ");
scanf("%s", str2);
// calculate the length of string str1
// and store it in integer variable
// you can also use strlen() library function to calculate the string length
for(i = 0; str1[i] != '\0'; i++);
for(j = 0; str2[j] != '\0'; j++, i++)
{
str1[i] = str2[j];
}
str1[i] = '\0';
// '\0' represents the end of string
printf("String After concatenation: %s", str1);
}
'; j++, i++)
    {
        str1[i] = str2[j];
    }

    str1[i] = '
```

```
// Created By Kaustav Ghosh Dastidar
#include <stdio.h>
void main()
{
char str1[50], str2[50], i, j;
printf("Enter first string: ");
scanf("%s", str1);
printf("Enter second string: ");
scanf("%s", str2);
// calculate the length of string str1
// and store it in integer variable
// you can also use strlen() library function to calculate the string length
for(i = 0; str1[i] != '\0'; i++);
for(j = 0; str2[j] != '\0'; j++, i++)
{
str1[i] = str2[j];
}
str1[i] = '\0';
// '\0' represents the end of string
printf("String After concatenation: %s", str1);
}
';
   // '
```

```
// Created By Kaustav Ghosh Dastidar
#include <stdio.h>
void main()
{
char str1[50], str2[50], i, j;
printf("Enter first string: ");
scanf("%s", str1);
printf("Enter second string: ");
scanf("%s", str2);
// calculate the length of string str1
// and store it in integer variable
// you can also use strlen() library function to calculate the string length
for(i = 0; str1[i] != '\0'; i++);
for(j = 0; str2[j] != '\0'; j++, i++)
{
str1[i] = str2[j];
}
str1[i] = '\0';
// '\0' represents the end of string
printf("String After concatenation: %s", str1);
}
' represents the end of string
   printf("String After concatenation: %s", str1);

}
```

Output

Enter first string: Gizmo

Enter second string:facts

String After concatenation: Gizmofacts

Next Steps?

Have any questions while reading the eBook? Want to discuss programming, gadgets and gaming? You can email us (kaustav@gizmofacts.com) where you can ask any questions you have in mind and subscribe to our Newsletter and visit our Facebook Page.

You can also check our blog "Gizmofacts" where you can find a ton of useful "how-to-guides", tutorials around blogging, programming, gadget reviews and gaming.