

www.gizmofacts.com

Learn Simple C Programs

Kaustav Ghosh Dastidar



GIZMOFACTS

Learn Simple C Programs

Introduction

Learn Simple C Programs is a very interesting introduction to the most commonly used programming language. It is an extremely user-friendly book for anyone looking to begin their journey into 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, 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

Copyright @ 2022 Kaustav Ghosh Dastidar All rights reserved.

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-Hello World	06
C Program-To Check If A Number Is Positive Or Negative	07
C Program-Convert Binary to Decimal	08
C Program-To Check Leap Year.....	9-10
C Program to print Fibonacci series in a given range.....	11
C Program to find factorial of a given number.....	12
C-Program To Display Twin Prime Number Between 1 To 100.....	13-14
C Program to check if given number is Armstrong or not.....	15-16
C Program to check if given number is palindrome or not.....	17-18
C Program-To Check If Number Is Even Or Odd.....	19
C Program to check whether an alphabet is vowel or consonant.....	20-21
C Program-To Print Pyramid Using Numbers.....	22-23
C Program-Print Triangle Using (*) Star.....	24-25
C Program-Print Diamond Using Star (*)......	26-27
Next Steps?.....	28

C Program-Hello World

Code description

Here we are sharing C programs on various topics such as *strings, series, arrays, loops, geometrical figures, mathematical calculation, sorting & searching algorithms and many more*. We are aiming to provide you a good packaged solution under one umbrella. For all the possible C programming questions. which might you have faced in interviews or any other assignments or projects. If you are not able to find the desired solution, please drop us comments so that we can get back to you to with some more and more beautiful programs. Happy coding.!!

This is simple C program where the message *Hello World* is been displayed using **printf()** funtion. The statement tells the compiler to include the standard input output file (**stdio.h**) in the program which contains the definitions of common input output functions such as **scanf()** and **printf()**. In the below program we are using **printf()** function.

The ‘main’ is the parent function of all the functions in C. We can use ‘void’ or ‘int’ as a return type. The keyword int is used for integer data types, so **int main()** always returns a value of integer data type. The function void **main()** does not have a return type or value.

The **printf()** function allows the content within double quotes to display on the screen

Source code

```
//Created by Kaustav Ghosh Dastidar.
#include <stdio.h>
int main()
//This main() function has ‘integer’ return type but you can also use ‘void’ for no return
{
    /* printf function displays the content that is
     * passed between the double quotes.
     */
    printf(“Hello World”);
    return 0;
}
```

Output

Hello World

C Program-To Check If A Number Is Positive Or Negative

Code description

Here we are sharing C programs on various topics such as *strings, series, arrays, loops, geometrical figures, mathematical calculation, sorting & searching algorithms and many more*. We are aiming to provide you a good packaged solution under one umbrella. For all the possible C programming questions. which might you have faced in interviews or any other assignments or projects. If you are not able to find the desired solution, please drop us comments so that we can get back to you to with some more and more beautiful programs. Happy coding.!!

Enter an integer you want to check as input. This integer is then stored in the variable 'number'. Now the program execution is done by checking whether this given integer is **greater or lesser than zero**. If it is greater than or equal to zero, then it is a positive number" otherwise it is a negative **number**

Source code

```
// Created by Kaustav Ghosh Dastidar.
#include <stdio.h>
int main()
{
    int number;

    printf("Enter a number \n");
    scanf("%d", &number);
    if (number >= 0)
        printf("%d is a positive number \n", number);
    else
        printf("%d is a negative number \n", number);
    return 0;
}
```

Output

Enter a number 9

9 is a positive number

C Program- Convert Binary to Decimal

Code Description

To obtain a decimal number, each digit of binary number need to be multiplied with power of 2 and then it is to be added with each multiplication result. The power starts from 0 and goes to n-1 where n is the total number of digits in binary number.

Source code

```
//Created by Kaustav Ghosh Dastidar.
#include<stdio.h>
#include<math.h>

int main()
{
    long int i,n,b=0,a;
    printf("The Binary number is: ");
    scanf("%ld",&n);
    printf("\nThe decimal equivalent of %ld is ",n);

    for(i=0;n!=0;++i)ww
    {
        a=n%10;
        b=(a)*(pow(2,i))+b;
        n=n/10;
    }

    printf("%ld",b);

    return 0;
}
```

Output

The Binary number is: 1111

The decimal equivalent of 1111 is 15

C Program- To Check Leap Year

Code description

In this program it is checked that whether the year entered by the user is firstly divided by 4.

*If it is divisible by 4 then it is divided by 100 and then 400.

*Now, If the year is divisible by all 3 numbers then that year is a leap year.

*If the year is divisible by 4 and 100 but not by 400 then it is not a leap year.

*If the year is divisible by 4 but not by 100, then it is a leap year.

(**Note:** if the year is divisible by 4 and not by hundred then the program does not check the last condition, i.e., whether the year is divisible by 400).

*If the year is not divisible by 4 then no other conditions are checked and the year is not a leap year.

Source code

```
// Created by Kaustav Ghosh Dastidar.
#include <stdio.h>
int main()
{
    int year;

    printf("Enter a year: ");
    scanf("%d",&year);

    if(year%4 == 0)
    {
        if( year%100 == 0)
        {
            // year is divisible by 400, hence the year is a leap year
            if ( year%400 == 0)
                printf("%d is a leap year.", year);
            else
                printf("%d is not a leap year.", year);
        }
    }
```

```
else
printf(“%d is a leap year.”, year );
}
else
printf(“%d is not a leap year.”, year);
ww
return 0;
}
```

Output

Enter a year: 2018

2018 is not a leap year.

C Program to print Fibonacci series in a given range

Code description

The Fibonacci sequence is a series where the next term is the **sum of pervious two terms**. The first two terms of the Fibonacci sequence is 0 followed by 1

Source code

```
// Created by Kaustav Ghosh Dastidar.
#include <stdio.h>
int main()
{
    int t1 = 0, t2 = 1, nextTerm = 0, n;

    printf("Enter a positive number: ");
    scanf("%d", &n);

    // displays the first two terms which is always 0 and 1
    printf("Fibonacci Series: %d, %d, ", t1, t2);

    nextTerm = t1 + t2;

    while(nextTerm <= n)
    {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }

    return 0;
}
```

Output

Enter a positive number: 99

Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,

C Program-To Find The Factorial Of A Number

Code description

This C program calculates the factorial of a given **positive number**. When any negative number is entered, the program displays error message. The type of factorial variable is declared as '**unsigned long long**' as the factorial of a number may be very large.

Source code

```
// Created by Kaustav Ghosh Dastidar.
#include <stdio.h>
int main()
{
    int n, i;
    unsigned long long factorial = 1;

    printf("Enter an integer: ");
    scanf("%d",&n);

    // show error if the user enters a negative integer
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");

    else
    {
        for(i=1; i<=n; ++i)
        {
            factorial *= i;          // factorial = factorial*i;
        }
        printf("Factorial of %d = %llu", n, factorial);
    }

    return 0;
}
```

Output

```
Enter an integer: 15
Factorial of 15 = 1307674368000
```

C-Program To Display Twin Prime Number Between 1 To 100

Code description

This is a simple “*C Program*” which will display twin prime numbers between *1 to 100*. We have also used a user-defined function to calculate the prime numbers.

Source code

```
// Created by Kaustav Ghosh Dastidar.  
// Created By Kaustav Ghosh Dastidar  
  
#include<stdio.h>  
#include<string.h>  
  
int prime(int);  
int main()  
{  
    int i,count=0;  
    printf("\n\n The twin primes between 1 to 100 :\n\n");  
    for(i=3;i<98;i=i+2)  
    {  
        if(prime(i)&&prime(i+2))  
        {  
            printf("< %d , %d >\n\n",i,i+2);  
            count++;  
        }  
    }  
    printf("\n\n No of twin prime pair : %d",count);  
    return 0;  
}  
int prime(int number)  
{  
    int j;  
    for(j=2;j<=number/2;j++)
```

```
        if((number%j)==0)
            break;
    if(j>number/2)
        return(1);
    return(0);
}
```

Output

The twin primes between 1 to 100 :< 3 , 5 >

< 5 , 7 >

< 11 , 13 >

< 17 , 19 >

< 29 , 31 >

< 41 , 43 >

< 59 , 61 >

< 71 , 73 >

No of twin prime pair : 8

C Program to check if given number is Armstrong or not

Code description

The number which is **equal to the sum of cubes of its digits** is known as **Armstrong number** . For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

Example:

$$371 = (3*3*3)+(7*7*7)+(1*1*1)$$

where:

$$(3*3*3)=27$$

$$(7*7*7)=343$$

$$(1*1*1)=1$$

So:

$$27+343+1=371$$

$$153 = (1*1*1)+(5*5*5)+(3*3*3)$$

where:

$$(1*1*1)=1$$

$$(5*5*5)=125$$

$$(3*3*3)=27$$

So:

$$1+125+27=153$$

User input value is copied into a **temporary variable named actualNumber**.

The last digit of the input value is extracted, cubed and added to the **result** variable.

The last digit is extracted till the **input value is not equal to 0**.

Now, the value of **result** variable is compared with the value of **actualNumber** variable.

If result and actualNumber are equivalent, then the input number is an Armstrong Number, else it is not.

Source code

```
#include <stdio.h>
// Created by Kaustav Ghosh Dastidar.
int main()
{
    int number, actualNumber, rem, result = 0;
    printf("Enter a three digit number: ");
    scanf("%d", &number);

    actualNumber = number;

    while (actualNumber != 0)
    {
        rem = actualNumber%10;
        result = result+( rem*rem*rem);
        actualNumber /= 10;
    }

    if(result == number)
        printf("%d is an Armstrong number.",number);
    else
        printf("%d is not an Armstrong number.",number);

    return 0;
}
```

Output1

Enter a three digit integer: 371

371 is an Armstrong number.

Output2

Enter a three digit integer: 135

135 is not an Armstrong number.

C Program-To Check Palindrome Number

Code description

This C program uses a **while loop** to reverse an integer which is entered by the user. Then, a **if statement** is used to check whether the reversed number is equal to the original number or not.

Atfirst the user enters **the originalNumber=101**.

originalNumber is stored in the variable **n**.

reversedNumber has been **initialised to 0**.

Now the while loop is executed (the condition $n > 0$ is satisfied).

calculate remainder:

remainder of 101, when divided by 10 $= (101 \% 10) = 1$

This 1 is stored in '*remainder*'.

now **reversedNumber** $= (\text{reversedNumber} * 10) + \text{remainder}$

$= (0 * 10) + 1$ (*we have initialized reverse=0*)

$= 1$

number $= \text{number} / 10$

$= 101 / 10$

$= 10$

Now the number is 10, the condition ($n > 0$) is satisfied. The above process is repeated for **number=10**.

remainder $= 10 \% 10 = 0$;

reverse $= (1 * 10) + 0 = 10$;

number $= 10 / 10 = 1$;

Now the number is 1, the condition ($n > 0$) is satisfied. The above process is repeated for **number=1**.

remainder $= 1 \% 10 = 1$;

reverse $= (10 * 10) + 1 = 101$;

number $= 1 / 10$ the condition $n > 0$ is not satisfied, control leaves the while loop.

The program ends here. Thus the given number is a palindrome number. As given number=101 equals the reverse of the number.

Source code

```
// Created by Kaustav Ghosh Dastidar.
```

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

```
int main()
```

```
{
```

```
    int n, reversedNumber = 0, remainder, originalNumber;
```

```
    printf("Enter an number: ");
```

```
    scanf("%d", &n);
```

```
    originalnumber = n;
```

```
    while( n>0 )
```

```
    {
```

```
        remainder = n%10;
```

```
        reversedNumber = reversedNumber*10 + remainder;
```

```
        n /= 10;
```

```
    }
```

```
    // Checks if originalNumber and reversedNumber are equal then it is a palindrome number.
```

```
    if (originalNumber== reversedNumber)
```

```
        printf("%d is a palindrome.", originalNumber);
```

```
    else
```

```
        printf("%d is not a palindrome.", originalNumber);
```

```
    return 0;
```

```
}
```

Output

```
Enter an integer: 101
```

```
101 is a palindrome.
```

C Program-To Check If Number Is Even Or Odd

Code description

At first the integer entered by the user is stored in variable **num**.

Now, using the modulus operator it has been checked whether the number is perfectly **divisible by 2 or not**

If the test expression **num%2 == 0 evaluates to 1 (true)** i.e the number is perfectly divisible by 2. So the number is even.

If the test expression **evaluates to 0 (false)** i.e the number is not divisible by 2 then the number is odd.

Source code

```
// Created by Kaustav Ghosh Dastidar.
#include<stdio.h>
int main()
{
    // This variable is to store the input number
    int num;

    printf("Enter an integer: ");
    scanf("%d",&num);

    // Modulus (%) returns remainder
    if ( num%2 == 0 )
        printf("%d is an even number", num);
    else
        printf("%d is an odd number", num);

    return 0;
}
```

Output1

Enter an integer: 8
8 is an even number

Output2

Enter an integer: 5
5 is an odd number

C Program-To Check Vowel or Consonant

Code description

At first the user needs to enter the character and this character is stored in variable `c`.

The `isLowercaseVowel` evaluates to **1 (true)** if `c` is a lowercase vowel and **0 (false)** for any other character.

Similarly, `isUppercaseVowel` evaluates to **1(true)** if `c` is an **uppercase vowel** and **0 (false)** for any other character.

If both `isLowercaseVowel` and `isUppercaseVowel` is equal to **0**, the test expression evaluates to **0 (false)** and the entered character is a **consonant**.

However, if either `isLowercaseVowel` or `isUppercaseVowel` is **1 (true)**, the test expression evaluates to **1 (true)** and the entered character is a **vowel**.

Source code

```
//Created by Kaustav Ghosh Dastidar.
```

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

```
int main()
```

```
{
```

```
char c;
```

```
int isLowercaseVowel, isUppercaseVowel;
```

```
printf("Enter an alphabet: ");
```

```
scanf("%c",&c);
```

```
// evaluates to 1 (true) if c is a lowercase vowel
```

```
isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
```

```
// evaluates to 1 (true) if c is an uppercase vowel
```

```
isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
```

```
// evaluates to 1 (true) if either isLowercaseVowel or isUppercaseVowel is true
```

```
if (isLowercaseVowel || isUppercaseVowel)
printf(“%c is a vowel.”, c);
else
printf(“%c is a consonant.”, c);
return 0;
}
```

Output1

Enter an alphabet: b

b is a consonant.

Output2

Enter an alphabet: A

A is a vowel.

C Program-To Print Pyramid Using Numbers

Code description

This C program will print a **right-angle Floyd's triangle** using numbers.

The numbers of rows need to be printed is stored in the integer variable '**num**'. As in this program we are looping through rows and columns we need two more integer variable '**row**' and '**column**'.

for (row = 1; row <= num; row++)- This is the outer for loop, which provides the access to each row. When the program will run this for loop will start from 1 then it becomes **2,3,4,5.....**(upto the number of rows entered by the user)

for (column = 1; column <= row; column++) – Now to get access of columns for each row we use this inner for loop. Every row has different no of columns. This program will start printing the pattern from the 1st columns of every row. Note: The **no of columns** to be printed is nothing but the **row number**.

printf(“%d\t”,row)-This print statement within the inner for loop will print the number of rows in each columns of the respected row. In this below example in the 7th row, the row no 7 is printed in each column of this row.

printf(“\n”)-After printing each row we need to move on to the next line. So this printf(“\n”) stament will print the new line.

Source code

```
// Created by Kaustav Ghosh Dastidar.
```

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

```
int main() {  
    int row, column;  
    int num;  
  
    printf(“Enter the number of rows :”);  
    scanf(“%d”, &num);  
    for (row = 1; row <= num; row++) {  
        for (column = 1; column <= row; column++) {
```

```
        printf("%d\t",row );
    }
    printf("\n");
}
return (0);
```

Output

Enter the number of rows :8

```
1
2   2
3   3   3
4   4   4   4
5   5   5   5   5
6   6   6   6   6   6
7   7   7   7   7   7   7
8   8   8   8   8   8   8   8
```

C Program-Print Triangle Using (*) Star

Code description

This C program will print the triangle using star.

The numbers of rows need to be printed is stored in the integer variable '**num**'. As in this program we are looping through rows and columns we need two more integer variable 'row' and 'column'.

for(i=1; i<=rows; ++i, k=0)-This is the outer for loop, which provides the access to each row. When the program will run this for loop will start from 1 then it becomes 2,3,4,5.....(upto the number of rows entered by the user)

for(space=1; space<=rows-i; ++space)- maximum number of rows-current rows

while(k != 2*i-1)-calculate the number of star to be printed

printf("* ")-prints the number of star

printf(" ")-prints number of space

printf("\n")-prints new line

Source code

```
#include <stdio.h>

int main()
{
    int i, space, rows, k=0;

    printf("Enter number of rows: ");
    scanf("%d",&rows);

    for(i=1; i<=rows; ++i, k=0)
    {
        for(space=1; space<=rows-i; ++space)
        {
            printf(" ");
        }

        while(k != 2*i-1)
```

```
    {  
        printf("* ");  
        ++k;  
    }  
    printf("\n");  
}  
return 0;  
}
```

Output

Enter number of rows: 5

```
    *  
  
  * * *  
  
* * * * *  
  
* * * * * * * *  
  
* * * * * * * * *
```

C Program-Print Diamond Using Star (*)

Code description

This C program is a combination of simple pyramid star pattern and inverted pyramid star pattern which consist of **$N*2-1$ rows**.

Each row contain spaces and stars which are printed in increasing and decreasing order.

Till Nth row stars are printed in increasing order. After Nth row stars are printed in decreasing order.

Till Nth row spaces are printed in decreasing order. After Nth row spaces are printed in increasing order.

Source code

```
// Created by Kaustav Ghosh Dastidar.
#include <stdio.h>
int main()
{
    int i, j, rows;
    int stars, spaces;

    printf("Enter rows to print : ");
    scanf("%d", &rows);

    stars = 1;
    spaces = rows - 1;

    /* Running a loop to iterate through rows */
    for(i=1; i<rows*2; i++)
    {
        /* Running a loop to Print spaces */
        for(j=1; j<=spaces; j++)
            printf(" ");

        /* Running a loop to Print stars */
        for(j=1; j<stars*2; j++)
            printf("*");
```

```

/* for next line */
printf("\n");

if(i<rows)
{

spaces--;
stars++;
}
}

return 0;
}

```

Output

Enter rows to print

: 5

```

    *

 ***

*****

*****

*****

*****

 ***

 *

```

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.