

## 1) Hello world program

```
#include <stdio.h>
int main()
{
    printf("I'm going to study tonight\n");
    int num;
    printf("Hello world!\n");
    printf("Coding is fun!\n");
    return 0;
}
```

### Output

study tonight

Hello world!

Coding is fun!

## 2) Program to take input of various datatypes in C

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

```
int main()
```

```
{
```

```
    printf("
```

```
int num1, num2;
```

```
float fraction;
```

```
char character;
```

```
    printf("Enter two numbers\n");
```

```
// taking integer input
```

```
scanf("%d %f", &num1, &num2);
```

```
    printf("\n\nThe two numbers you have
```

```
entered are %d and %f\n", num1, num2);
```

```
Result : num1 + num2 = 3.0 for num1 = 1.0 and num2 = 2.0
```

```

// Taking float as I/O
printf("In\nEnter a Decimal number\n");
scanf("%f", &fraction);
printf("In\nThe float or fraction that you have entered is %f", fraction);

// Taking character as input
printf("In\nEnter a character\n");
scanf("%c", &character);
printf("In\nThe character that you have entered is %c", character);

printf("In\nIt coding is fun!\n");
return 0;
}

```

### Output

The two numbers you have entered are 34 and 7

Enter a decimal number  
5.4

The float or fraction that you have entered is  
5.400000

Enter a character  
a

The character you have entered is a  
Coding is fun!

### ② ASCII value of character

```

#include<stdio.h>
int main()
{
    printf("In\nIt studytonight\n");
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);
    printf("In\nASCII value of %c = %d", c, c);
}

```

```

#include<stdio.h>
main()
{
    char ch;
    printf("Enter one character: ");
    scanf("%c", &ch);
}

```

```
return 0;  
}
```

```
printf("ASCII value is : %d", ch);
```

### Output

Enter a character : s  
ASCII value of s = 115

```
("#include<stdio.h>\nint main()\n{\n    char ch;\n    printf("Enter a character : ");\n    scanf("%c", &ch);\n    printf("ASCII value is : %d", ch);\n}\n\n")
```

### (A) program to use gets() function

Scansf() and gets() both are same which take the input from the user.

→ scanf() = can only take input until it encounters a space. The words after space are ignored by it.

→ gets() = is used to take a single input at a time but can be used to input a complete sentence with spaces unlike scanf().

```
#include<stdio.h>\nint main()\n{\n    char str[50];\n\n    printf("Enter your complete name : \n");\n    gets(str);\n\n    printf("Welcome to studytonight %s\n", str);\n    printf("Coding is fun\n");\n\n    return 0;\n}
```

### Output

Enter your complete name

Lavanya Somanna

Welcome to studytonight Lavanya Somanna  
Coding is fun!

```
("#include<stdio.h>\nint main()\n{\n    char str[50];\n\n    printf("Enter your complete name : \n");\n    gets(str);\n\n    printf("Welcome to studytonight %s\n", str);\n    printf("Coding is fun\n");\n\n    return 0;\n}\n\n")
```

5 Basic "if - else" condition

```
#include<stdio.h>
int main()
{
    printf("In Init Study Tonight");
    int number;
    printf("Please enter a number : ");
    scanf("%d", &number);

    if (number < 100)
        printf("Number is less than 100\n");
    else if (number == 100)
        printf("Number is 100\n");
    else
        printf("Number is greater than 100\n");
    return 0;
}
```

## ⑥ Switch Case with "break"

- switch() can only contain char and int.
- break is used to exit from switch statement.
- Char variable is always initialized within its own case.

```
#include<stdio.h>
int main()
{
    printf("Enter grade : ");
    char grade;
    printf("Enter your grade : ");
    scanf("%c", &grade);

    switch(grade)
    {
        case 'A':
        case 'B':
        case 'C':
            printf("Grade is %c", grade);
            break;
        case 'D':
            printf("Grade is %c", grade);
            break;
        case 'F':
            printf("Grade is %c", grade);
            break;
        default:
            printf("Grade is %c", grade);
    }
}
```

Case 'A':

```
printf("Excellent\n");
break;
```

Case 'B':

```
printf("Keep it up\n");
break;
```

Case 'C':

```
printf("well done\n");
break;
```

Case 'D':

```
printf("you passed\n");
break;
```

Case 'E':

```
printf("Better luck next time\n");
break;
```

default:

```
printf("Invalid grade\n");
```

}

```
printf("your grade is %c\n", grade);
```

```
return 0;
```

}

### Output

Enter your grade

c

well done

your grade is c

if (-)

if ( )

{ }

else

{ }

if ( )

{ printf(

"done , may win?

)

if ( )

{ if ( )

switch case without break

If there is no break statement then the cases following the matched case other than default will get executed

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

```
int main()
```

{

char grade;

```
printf("Enter your grade:\n");
```

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

```
switch (grade)
```

{

```

Case 'A':
    printf("Excellent\n");
Case 'B':
    printf("Inn Keep it up\n");
Case 'C':
    printf("In Case C: well done\n");
Case 'D':
    printf("In Case D: You failed\n");
Case 'F':
    printf("In Case F: Better luck next time\n");
default:
    printf("In default Case: Invalid grade\n");
}
printf("Your grade is %c\n", grade);
return 0;
}

```

### Output

Enter your grade : B

Keep it up!

No break statement hence all following cases executes

Case C: well done!

Case D: You failed!

Case F: Better luck next time

Default: Invalid case

## using switch case

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Input a character: ");
    scanf("%c", &ch);
    switch(ch)
    {
        case 'a':
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U':
        case 'U':
            printf("\n\n%c is a vowel.\n\n", ch);
            break;
        default:
            printf("\n\n%c is not a vowel.\n\n", ch);
    }
    return 0;
}
```

### Output

Input a character: a  
a is a vowel

## ⑧ program to reverse the case of input character

- getchar() is similar to scanf()
- islower() is system defined function under ctype.h header file to check if the character is in lowercase or not
- toupper(). Converts the input parameter into equivalent uppercase char
- putchar() is similar to printf()

```

#include <stdio.h>
#include <ctype.h>
int main()
{
    printf("In this program we will learn how to convert uppercase to lowercase and vice versa.\n");
    char alphabet;
    printf("Enter an alphabet : ");
    //putchar('\n');
    //alphabet = getchar();
    scanf("%c", &alphabet);
    printf("in Reverse case of %c is : ", alphabet);
    if (islower(alphabet))
        putchar(toupper(alphabet));
    else (isupper)
        printf("%c", tolower(alphabet));
    printf("In In Coding is fun\n");
    return 0;
}

```

### Output

Enter an alphabet : s

Reverse case of s is S

## Q) Swapping two numbers using Temporary variable

```

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

void main()
{
    int x=10, y=15, temp;
    temp = x;      temp = 10
    x = y;      x = 15
    y = temp;      y = 10
    printf("x = %d and y = %d", x, y);
    getch();
}

```

### Output

x=15 and y=10

## Swapping two numbers without using a temporary variable

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int x=10, y=15;
    x = x+y - (y=x); // condition d) 138
    printf("x=%d and y=%d", x, y);
    getch();
}
```

$$\begin{aligned} x &= x+y & x &= 25 \\ y &= x-y & y &= 25-15=10 \\ x &= x-y & x &= 25-10=15 \end{aligned}$$

### Output

x=15 and y=10

$$10+15-(10)$$

$$25-10=15=x$$

## Swapping two numbers using Bitwise operator

```
#include<stdio.h> 00 = 0
#include<conio.h> 01 = 1
void main()
{
    int x=6, y=4;
    x = x^y; x = 2;
    y = x^y;
    x = x^y;
    printf("x=%d and y=%d", x, y);
    getch();
}
```

$$\begin{array}{l} \text{Initial State: } x=0110, y=0100 \\ \text{Step 1: } x=0110 \oplus 0100 = 0010 \\ \text{Step 2: } y=0010 \oplus 0100 = 0110 \end{array}$$

### Output

x=4 and y=6

## Swapping two numbers using Multiplication & division

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int x=6, y=4;
    x = x*y; x = 24
    y = x/y; y = 6
}
```

```

x = x / y; // A division operation
printf("x = %d and y = %d", x, y);
getch();

```

## output

$$x = 4 \quad \text{and} \quad y = 6$$

16 Program to print the Largest and Smallest Using Global Declaration

Some important points about Global variable declarations are:

- It can be done anywhere within the program
  - unlike local variables that can be used within the scope of a particular function
  - it is used to assign the input value to the variable and store it at that particular location.
  - and is used to represent numbers in a digit format with leading 0's

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

```
int a,b;
```

fun main()

{  $(\psi(x) \wedge b \leq b \text{ and } b \neq x)$  } for all  $b$

```
printf("In Enter the two values to find greatest and  
smallest number: \n");  
scanf("%d %d", &a, &b);
```

if ( $a == b$ )

```
printf (" Both  are equal\n");
```

else if ( $a < b$ )

{

```
printf ("In\n The largest number is %03d\n", b);  
printf ("In\n The smallest number is %03d\n", a);  
printf ("In\n The largest number is %03d\n", b);
```

3

```
else {  
    if (b < a)  
    {  
        printf ("The largest number is %d\n", a);  
        printf ("The smallest number is %d\n", b);  
    }  
    return 0;  
}
```

## Output

Enter the two values to find gce & smallest no.

10

七

The largest number is 010

The smallest number is 007

100.01612284000000

→ Loop

## ① Basic "for" loop program

\* Initialization new print medium printing at 100%<sup>100%</sup>

## \* Condition

## \* ~~update~~ updaton

for (initialization , condition, increment)

{ Code statements;

۹

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

int main()

{ p o d o

```
for (i=0; i<10; i++)
```

```
{ printf ("i=%d\n", i);
```

3

Printf ("In the value of i after exiting the loop is %d\n", i);

return 0;

3  
((0, "Hello" is return type of int) first)  
((0, "Hello" is return value of int) first)

Output

i = 0  
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6  
i = 7  
i = 8  
i = 9

## ② Simple while loop program

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

```
int main()
```

```
{
```

```
    int i=0;
```

```
    printf("In printing numbers using while loop  
          from 0 to 9\n");
```

```
    while (i<10)
```

```
    {
```

```
        printf("%d\n", i);
```

```
        i++;
```

```
}
```

```
    return 0;
```

```
}
```

Output

printing numbers using while loop from 0 to 9

0

1

2

3

4

5

6

7

8

9

③ Basic do while loop program

→ DO While Loop is used when the actual code must be executed atleast once.

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

```
int main()
```

```
{
```

```
    int i=10;
```

```
    do
```

```
    { printf("i=%d\n", i);
```

```
        i = i-1;
```

```
}
```

```
    while (i>0)
```

```
    printf("The value of i after exiting the loop  
is %d\n", i);
```

```
    printf("Coding is fun");
```

```
    return 0;
```

```
}
```

### Output

```
i=10
```

```
i=9
```

```
i=8
```

```
i=7
```

```
i=6
```

```
i=5
```

```
i=4
```

```
i=3
```

```
i=2
```

```
i=1
```

```
i=0
```

The value after exiting the loop is 0

- ④ Basic program to show use of nested loops
- \* for Loops are very good when we have to do some work which requires repetition.
  - used to print a pattern in c.
  - used to print out matrix using a 2 dimensional array
  - a lot of other patterns like pyramid of numbers etc.

\* Using a loop inside another loop is called nested loop

```
#include <stdio.h>
int main()
{
    int i, j, k;
    printf ("The output of the nested loop is :\n\n");
    for (i=0; i<5; i++)
    {
        printf ("%d %d %d\n", i, i+1, i+2);
        for (j=0; j<5; j++)
        {
            printf ("* ");
        }
        printf ("\n");
    }
    printf ("");
    return 0;
}
```

### Output

Output of the nested loop is: ✓

```
5 6 7
5 6 7
5 6 7
5 6 7
5 6 7
```

## ⑤ program to print factorial of a number

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int fact, i, n;
    fact = 1;
    printf ("Enter the number \t");
    scanf ("%d", &n);
    for (i=1; i<=n; i++)
    {
        fact = fact * i;
    }
    printf ("Factorial of %d is %d", n, fact);
    getch();
}
```

### Output

Enter the number 5

Factorial of 5 is 120

## ⑥ program to print the Fibonacci Series starting from 0

```
#include <stdio.h>
#include <conio.h>
void fibonacci(int num);
void main()
{
    int num=0;
    clrscr();
    printf ("Enter the number of terms \t");
    scanf ("%d", &num);
    fibonacci (num);
    getch();
}
```

```
void fibonacci(int num)
```

```
{  
    int a, b, c, i = 3;  
    a = 0;  
    b = 1;  
    if (num == 1)  
        printf("%d", a);  
    if (num >= 2)  
        printf("%d", a, b);  
    if  
    while (i <= num)  
    {  
        c = a + b;  
        printf("%d", c);  
        a = b;  
        b = c;  
        i++;  
    }  
}
```

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

```
int main()  
{  
    int n, a=0, b=1, c, i;  
    printf("Enter the limit");  
    scanf("%d", &n);  
    for (i=1; i<=n; i++)
```

```
{  
    printf("%d", a);  
    c = a + b;  
    a = b;  
    b = c;  
}
```

3      1      2      3      5  
1st      2nd

$a = 0, b = 1$        $c = 1$   
 $a = 1, b = 1$        $c = 2$   
 $a = 1, b = 2$        $c = 3$   
 $a = 2, b = 3$        $c = 5$   
in a value      8      8      13  
2. decimal      21      21

### Output

Enter number of terms 6

0 1 1 2 3 5

0 1 2 3 5 8

⑦ program to check whether a Number is a Palindrome

A Palindrome is a number or a string which is similar when read from the front and from the rear. ~~ex- 121~~ or oppo etc.

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

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

```
void main()
```

```
{  
    // won't do return int main() function  
    int a, b, c, s = 0;  
    clrscr();  
    printf("Enter a number : ");
```

0 = main fun

( ) return

(main) = b x

(main) = 121

( ) return

scanf("%d", &n);

121

C = 1; // It will go inside the loop  
// no. is reversed inside the while loop  
while (n >= 0) {  
 // exit from loop

{

R = n % 10; // 1 2 1

S = (S \* 10) + R; // 12 121

n = n / 10; // 12 1 0

}

// here the reversed number is compared with given no.

if (S == C)

{ printf("The number %d is a palindrome", C);

}

else

{ printf("The number %d is not a palindrome", C);

}

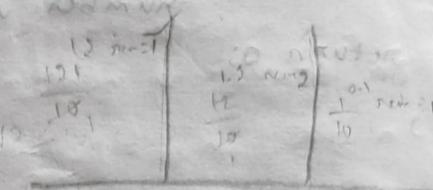
getch();

}

Output

Enter the number : 121

The number 121 is a palindrome



$$121 = \frac{1}{10} \Rightarrow 12 \text{ rem } 1$$

12 rem 1

```

    printf("In In sum of the digits of the entered
           number is = %d\n", sum);
    return 0;
}

```

### Output

Enter the number you want to add digits of : 5467  
 Sum of the digits of the entered number is = 22

in coming pages

### Q) Program to reverse a string

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int i, j, k;
    char str[100];
    char rev[100];
    printf("Enter a string : ");
    scanf("%s", str);
    printf("The original string is %s\n", str);
    for (i=0; str[i]!='\0'; i++);
    {
        k = i-1;
    }
    for (j=0; j<=i-1; j++)
    {
        rev[j] = str[k];
        k--;
    }
    printf("The reverse string is %s\n", rev);
    getch();
}

```

### Output

Enter a string : studytonight

The original string is studytonight  
 The reverse string is thgintoydnuts

## ⇒ NUMBERS

① program to find average of n numbers

```
#include<stdio.h>
int main()
{
    int n, i;
    float sum = 0, x;
    printf("Enter number of elements : ");
    scanf("%d", &n);
    printf("Enter %d element (%d", n);
    for (i=0; i<n; i++)
    {
        scanf("%f", &x);
        sum += x; → sum = sum + x
    }
    printf("Average of the entered numbers is = %.2f", (sum/n));
    return 0;
}
```

Output

Enter number of element : 5

Enter 5 elements

3

5

6

7

2

Average of the entered numbers is = 4.600000

② program to find Armstrong number  
b/w 1 to 500

An Armstrong number or Narcissistic number is a n digit number such that the sum of its digits raised to the nth power is equal to the number itself.

For ex Let's take an armstrong number: 153, which is 3 digit number, here  $1^3 + 5^3 + 3^3 = 1 + 125 + 27$  which equal to 153

```
#include<stdio.h>
#include<math.h>
int main()
```

```
{
```

int n, sum, i, C, R;  
printf("The Armstrong numbers in b/w 1 to 500  
are: %n %n %n");

```
for (i=1; i<=500; i++)
```

```
{
```

C = i; 1 2 153  
sum = 0;  
while (i != 0)

```
{
```

R = i % 10; 1 2 3

sum += R \* R \* R; 1 8

i = i / 10; 0.1 0.2 15

if (sum == C)

```
printf("Init %d", i);
```

```
return 0;
```

```
}
```

$$t = 153$$

$$a = t \% 10 \rightarrow ③$$

$$\text{sum} = \text{sum} + a * a * a$$

$$= 0 + 27 \rightarrow ④$$

$$= 27$$

$$=$$

## Output

The Armstrong numbers in b/w 1 to 500 is

1

153

370

371

407

## Program to check whether a number is Armstrong number

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    int n, sum=0, t, a;
```

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

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

```
    t = n;
```

```
    while(n != 0)
```

```
{
```

```
    a = n % 10;
```

```
    sum += a * a * a;
```

```
    n = n / 10;
```

```
}
```

```
    printf("Initial sum = %d", sum);
```

```
    if (sum == t)
```

```
        printf("%d is an Armstrong number\n", t);
```

```
    else
```

```
        printf("%d is not an Armstrong number\n", t);
```

```
    return 0;
```

```
}
```

## Output

Enter a number : 371

sum = 371

371 is an Armstrong number

371

a=n%10

sum+=a\*a\*a

n=n/10

t+=a\*a\*a

n=n/10

### ③ Checking for odd and Even numbers, using Bitwise operator

→  $x \& 1$  returns true if the LSB (Least significant bit) of binary representation of an integer  $x$  is 1. It returns false if the LSB or the Right most bit in a binary sequence is 0.

In binary representation of a integer, if LSB is 1 then it is odd and if LSB is 0 then it is even.

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

```
int main()
{
    int x;
    for (x=0; x<=10; x++)
    {
        if (x&1) // if number is odd
            printf("It's %d is odd\n", x);
        else if (! (x&1))
            printf("It's %d is even\n", x);
    }
    return 0;
}
```

#### Output

```
0 is even
1 is odd
2 is even
3 is odd
4 is even
5 is odd
6 is even
7 is odd
8 is even
9 is odd
10 is even
```

Don't  
forget  
that  
if  
you  
want  
to  
check  
if  
a  
number  
is  
odd  
or  
even  
then  
you  
have  
to  
check  
the  
LSB  
bit  
which  
is  
the  
right  
most  
bit  
of  
the  
number  
so  
if  
it  
is  
1  
then  
it  
is  
odd  
else  
it  
is  
even

```
#include <stdio.h>
int main()
{
    char ch;
    scanf("%c", &ch);
    if ((ch >='a' && ch <='z') || (ch >='A'
        && ch <='Z'))
        { If ("'x.c'm alpha"; ch); }
    else if (ch >='0' && ch <='9')
        { If ("'y.d'm digit"; ch); }
    else { If ("'z.'m special char"; ch); }
    return 0;
}
```

Checking if given number is odd or even  
number w/o using % (Mod) operator

```
#include<stdio.h>
int main()
{
    printf("Enter a number: ");
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    if ((n/2)*2 == n)
        printf("The number is even (%d)", n);
    else
        printf("The number is odd (%d)", n);
    return 0;
}
```

### Output

Enter a number: 23

23 is odd

### ④ program to find Factors of a Number {

```
#include<stdio.h>
int main()
{
    printf("Enter the number to find the factors of: ");
    scanf("%d", &num);
    printf("The factors of %d are ", num);
    for (i=1; i<=num/2; i++)
    {
        if (num % i == 0)
            printf("%d ", i);
    }
    return 0;
}
```

Enter the number to find the factors of : 26

Factors of 26 are : 1  
2  
13

### ⑤ program to find sum of N input numbers in C

```
#include <stdio.h>
int main()
{
    int n, sum=0, c, value;
    printf("nEnter the number of integers you want to add:");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);
    for (c=1; c<=n; c++)
    {
        scanf("%d", &value);
        sum += value;
    }
    printf("nsum of entered numbers = %d", sum);
    printf("n coding is fun");
    return 0;
}
```

#### output

Enter the number of integers you want to add: 5  
Enter 5 integers  
5  
6  
2  
4  
1

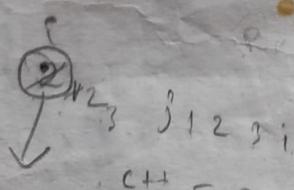
sum of entered numbers = 18

⑥ program to find first N prime numbers.

```
#include <stdio.h>
int main()
{
    int n, i=3, count, c;
    printf("Enter the number of prime numbers required : ");
    scanf("%d", &n);
    if (n >= 1)
    {
        printf("The first %d prime numbers are : ", n);
        printf("2");
        for (i=2; i<n; i++)
        {
            for (c=2; c<i; c++)
            {
                if (i%c == 0)
                    break;
            }
            if (c==i)
                printf("\n%d", i);
            count++;
        }
        return 0;
    }
}
```

Output

Enter the number of prime numbers required : 7  
 First 7 prime numbers are : 2 3 5 7 11 13 17



i--  
 c = c - 1

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, n;
    printf("Enter N value : ");
    scanf("%d", &n);
    int c=0, sum=0;
    for (i=1; i<=n; i++)
    {
        if (c==0)
        {
            for (j=1; j<=i; j++)
            {
                if (i%j==0)
                    c++;
            }
            if (c==2)
                printf("%d\n", i);
            sum += i;
            c=0;
        }
    }
    printf("Sum = %d", sum);
}
```