

Function: printf()

library: **<stdio.h>**

sends the formatted data to the standard output stream (stdout)

```
int printf ( format_string , argument_1 , argument_2 , . . . ) ;
```

format string → it is a string constant (in quotes) containing:

- ordinary characters (which are simply copied to the screen)
- formatting codes of given arguments:

- %c** – a single character
- %s** – string (array of characters)
- %d** – signed decimal number
- %f** – floating-point number (decimal notation)
- %e** – floating-point (scientific notation: mantissa/exponent)
- %g** – floating-point (the shortest representation: %e or %f %)
- %u** – unsigned decimal integer
- %x** – unsigned hexadecimal integer
- %o** – unsigned octal integer
- h l L** – length subspecifiers (short int / long int / long double)

example:

```
#include <stdio.h>                                     // Note: this lecture, most examples
int main( )                                           // are implemented in “C” programming language
{
    int x = 10;
    long y = 20;
    double s;
    s = x + y;
    printf ( "%"s of calculation: %d + %ld = %f" , "Result" , x , y , s );
}
```

effect on the screen → **Result of calculation: 10 + 20 = 30.000000**

length codes (with and precision subspecifiers) can be added to determine the number of digits that are printed: **%Xd** **%X.Xf**

- e.g. **%4d** – decimal number at 4 positions
- %10f** – floating (real) number at 10 positions
- %10.2f** – floating at 10 positions with 2 digits after decimal point
- %.3f** – floating-point number with 3 digits after the decimal point

Function: scanf()

<stdio.h>

reads the data from the standard input stream (stdin), according to the specified format and stores them under the specified memory addresses

```
int scanf ( format_string , pointer_1 , pointer_2 , . . . ) ;
```

format_string → it is a string constant (in quotes) containing instructions how to treat the next data loading from stream (what types of variables are at addresses *pointer_1*, *pointer_2* ...)

Formatting codes, similar to the printf ():

- %c** – a single character
- %s** – array of characters - string
- %d** – signed integer
- %f** lub **%e** – floating-point number
- %u** – unsigned integer
- %x** – hexadecimal integer
- %o** – octal integer
- h** **l** – length specifier for: **d u x o** (short or long int)
- l** – length specifier for: **f e** (double)
- L** – length specifier for: **f e** (long double)

& – reference / address operator (returns the address of the variable)

example:

```
#include <stdio.h> // implementation in "C"
int main( )
{
    int x;
    double y;
    char letter;
    printf( "Enter an integer number: " );
    scanf ( "%d" , &x );
    printf( " Specify the one real number and one character: ");
    scanf ( "%lf %c" , &y , &letter );
}
```

Printout → **Enter an integer number:**

Reading ← **123** ↵

Printout → **Specify the one real number and one character:**

Reading ← **456.789 a** ↵

The result of reading: **x == 123**, **y == 456.789**, **letter == 'a'**

BASIC INSTRUCTIONS OF C / C ++

- grouping instruction - braces { } are used to group a number of declarations and statements in a compound statement (one block).

example:

```
#include <stdio.h>
int main( )
{
    int a = 10, b = 20 ;
    {
        int a = 30 ;           // 'override' the previous definition of variable 'a'
        printf( "A = %d, B = %d \n" , a , b );           // printout:  A=30, B=20
    }
    printf( "A = %d, B = %d \n" , a , b );           // printout:  A=10, B=20
    . . .
    if( a > 0 )
    {
        printf( " Enter the new value of A =" );
        scanf( "%d" , &a );
    }
}
```

- conditional statement (can be one of two forms):

```
if ( expression )
    internal_instruction ;
```

this instruction checks if the expression is true (a nonzero value)

e.e.. if (*expression*) is equivalent to if (*expression* != 0)

example:

```
#include <stdio.h>                                     // implementation in "C"
int main( )
{
    int number;
    printf( "Specify any integer A =" ); scanf( "%d" , &number );
    if( number % 2 == 0 )                               // if the remainder of the division by 2 is 0
        printf( " The number is even" );
}
```

or complex form of conditional statement (with else):

```
if ( expression )
    instruction_1 ;
else
    instruction_2 ;
```

example:

```
#include <stdio.h> // implementation in "C"
int main( )
{
    char L;
    printf( "Give any a capital letter L = " );    scanf( "%c" , &L );
    if( L >= 'A' && L <= 'Z' )
        printf( "\n\n Well done! It is a capital letter" );
    else
        printf( "\n\n Wrong! This is NOT a capital" );
    printf( "\n\n Press ENTER to complete the program" );
    fflush(stdin); // clearing the stream buffer <stdin> , here the keyboard
    getchar( );
}
```

Further examples for the conditional statement:

```
#include <stdio.h>

int main( ) // "C", the maximum value from the three numbers
{
    int A, B, C;
    printf( "Enter the first number: " );    scanf( "%d" , &A );
    printf( "Enter a second number: " );    scanf( "%d" , &B );
    printf( "Enter a third number: " );    scanf( "%d" , &C );

    if( A > B && A > C ) printf( "Maximum = %d" , A );
    if( B > A && B > C ) printf( "Maximum = %d" , B );
    if( C > A && C > B ) printf( "Maximum = %d" , C );

    printf( "\n\n Press ENTER to complete the program" );
    fflush(stdin); getchar();
}

// What happens when two or three numbers are equal ?
```

Other versions of the same (maximum) program

```
include <stdio.h>
int main()                                     // “C”, The maximum value from the three numbers
{
    int A, B, C;
    printf( "Enter the first number: " );      scanf( "%d" , &A );
    printf( "Enter a second number: " );      scanf( "%d" , &B );
    printf( "Enter a third number: " );       scanf( "%d" , &C );

    if( A > B )
        if( A > C )
            printf( "Maximum = %d" , A );
        else
            printf( "Maximum = %d" , C );
    else
        if( B > C )
            printf( "Maximum = %d" , B );
        else
            printf( "Maximum = %d" , C );

    fflush(stdin); getchar();
}
```

```
#include <stdio.h>                             // “C”, The maximum value from the three numbers
int main( )
{
    int A, B, C, max;
    printf( "Enter the first number: " );
    scanf( "%d" , &A );
    printf( "Enter a second number: " );
    scanf( "%d" , &B );
    printf( "Enter a third number: " );
    scanf( "%d" , &C );

    max = A;
    if( max < B ) max = B;
    if( max < C ) max = C;

    printf( "\n The maximum value = %d" , max );
    fflush(stdin); getchar();
}
```

```
/* Program solving quadratic / trinomial equation  $Ax^2+Bx+C=0$ 
as an illustration of nesting conditional statements */
```

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

```
// implementation in "C"
```

```
#define _USE_MATH_DEFINES
```

```
// in <math.h> Microsoft library
```

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

```
// a mathematical function of the square root: sqrt
```

```
int main( )
```

```
{
```

```
    double a, b, c, delta, x1, x2;
```

```
    printf( "Enter the first coefficient A= " );
```

```
    scanf( "%lf" , &a );
```

```
// Caution !!! %lf not a %f
```

```
    printf( " Enter the second B= " );
```

```
    scanf( "%lf" , &b );
```

```
    printf( "Enter the third C= " );
```

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

```
    delta = b*b - 4*a*c;
```

```
    if( delta < 0 )
```

```
        printf( "\n No solutions" );
```

```
    else
```

```
        if( delta == 0 )
```

```
        {
```

```
            x1 = x2 = -b/(2*a);
```

```
            printf( "There is one solution: x1=x2= %f", x1 );
```

```
        }
```

```
    else
```

```
    {
```

```
        x1 = (-b - sqrt(delta)) / (2*a);  x2 = (-b + sqrt(delta)) / (2*a);
```

```
        printf( "There are two solutions: x1= %.2f, x2= %.2f", x1, x2 );
```

```
    }
```

```
    printf( "\n\n Press ENTER to complete the program" );
```

```
    fflush(stdin);
```

```
    getchar();
```

```
}
```

- Nested else-if:

```

if ( condition_1 )
    statement_1;
else
if ( condition_2 )
    statement_2;
else
if ( condition_3 )
    statement_3;
else
    default_statement_4;

```

- Multiple selection:

```

switch ( control_expression )
{
    case value_1 :    statement_1;
                    break;

    case value_2 :
    case value_3 :
    case value_4 :    statement_234;
                    break;

                    default : default_statement;
                    break;
}

```

```

#include <stdio.h>
int main( )
{
    int number;
    printf( " Specify an integer value A =" );
    scanf( "%d" , & number );
    switch( number )
    {
        case 0 : printf( "You have entered the zero" ); break;
        case -5 : printf( "You have entered minus five" ); break;
        case 7 : printf( "You have entered the number seven" );
                break;
        case 9 : printf( "You have entered the number nine" ); break;
        default: printf( " A number different than: 0, -5, 7, 9 " );
                break;
    }
}

```

// implementation in "C"

An example of using the **switch** statement:

```
#include <stdio.h>                                     // „C”, Simple calculator program with a "menu"

int main( )
{
    char selected_option;
    double a, b, result;

    printf( "Enter the first number A = " );           // read two numbers from the keyboard
    scanf( "%lf" , &a );
    printf( " Enter the second number B = " );
    scanf( "%lf" , &b );

    printf( "\n\nPossible operations:" );           // display the "menu" of possible options
    printf( "\n (+) result = A + B" );
    printf( "\n (-) result = A - B" );
    printf( "\n (*) result = A * B" );
    printf( "\n (/) result = A / B" );
    printf( "\n\nEnter the symbol of the operation: " );

    fflush(stdin);

    selected_option = getchar( ); // read the ASCII character of the selected operation

    switch( selected_option ) // switch to select one of the arithmetic operations
    {
        case '+': result = a + b; break;
        case '-': result = a - b; break;
        case '*': result = a * b;
                break;
        case '/': result = a / b; break;
        default: result = 0;
                printf( "\n Error: bad operation specified \n" );
                break;
    }

    // printout of input numbers and final result, with a given accuracy of decimal places
    printf( "\nFinal result of calculations: %.1f %c %.1f = %.2f ",
           a , selected_option , b , result );

    printf( "\n\n Press ENTER to complete the program " );
    fflush( stdin );
    getchar( );
}
```