

Input in C

- **Reading in value of the built-in data types**

- The `scanf()` function is used to read in **values** of **all built-in data types** in C.

- **Syntax** of the `printf()` function:

```
scanf ( " format string " , &var1, &var2, ..... );
```

Very important note:

- The `&` character is the **"reference" operator** of the C programming language
- The expression `&x` means: the **address** of the variable `x`
- You **must** pass the **address** of a variable to the `scanf()` function for reading operations.

We will discuss the `&` operator in more details later....

- **Format string:**

- The **format string** in the `scanf()` function contains **formatting characters** that instruct the **C compiler** to **read in a value** and **store** it in the given **representation (encoding)**

- **Commonly used formatting characters:**

Formatting character	Meaning
<code>%d</code>	Print the (next) value as a signed integer value
<code>%u</code>	Print the (next) value as a unsigned integer value
<code>%ld</code>	Print the (next) value as a long signed integer value
<code>%lu</code>	Print the (next) value as a long unsigned integer value
<code>%f</code>	Print the (next) value as a floating point value
<code>%lf</code>	Print the (next) value as a double precision floating point value

<code>%c</code>	Print the (next) value as a character (i.e., used the ASCII code)
<code>%s</code>	Print the (next) value as a string (will be explained much later)

○ **Example:**

```
int main( int argc, char* argv[] )
{
    int i;
    float x;

    printf( "Enter an integer value:" );
    scanf( "%d", &i );
    printf( "i = %d\n", i );

    printf( "Enter a floating point value:" );
    scanf( "%f", &x );
    printf( "x = %f\n", x );
}
```

Sample Output:

```
Enter an integer value:56
i = 56

Enter a floating point value:3.14
x = 3.140000
```

○ **Example Program:** (Demo above code)

Example

- Prog file: [click here](#)

How to run the program:

- **Right click** on link and **save** in a scratch directory
- To compile: `gcc scanf1.c`
- To run: `./a.out`

• **Reading a serie of inputs**

○ **Fact:**

- `scanf()` return the **number of items** read
- `scanf()` return the value **-1** when you type **control-D** at the keyboard
(The character **control-D** represents **end-of-file (EOF)** in UNIX)

- **How to** read a serie of **integers** until **EOF**:

```
int main( int argc, char* argv[] )
{
    int i;

    printf( "Enter an integer value: " );

    while ( scanf( "%d", &i ) != EOF )
    {
        printf( "i = %d\n", i );
        printf( "Enter another integer value (type ^D to end): " );
    }

    printf( "\nDONE !\n" );
}
```

- **Example Program:** (Demo above code)

Example

- Prog file: [click here](#)

How to run the program:

- **Right click** on link and **save** in a scratch directory
- To compile: `gcc scanf2.c`
- To run: `./a.out`