

Programming in C++

Tushar B. Kute,
<http://tusharkute.com>



Array

- An array is defined as the collection of similar type of data items stored at contiguous memory locations.
- Arrays are the derived data type in C++ programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc.
- The array is the simplest data structure where each data element can be randomly accessed by using its index number.

Array

- We can declare an array in the c language in the following way.

```
data_type array_name[array_size];
```

- Now, let us see the example to declare the array.

```
int marks[5];
```

- Here, int is the data_type, marks are the array_name, and 5 is the array_size.

Strings

- The string can be defined as the one-dimensional array of characters terminated by a null ('\0').
- The character array or the string is used to manipulate text such as word or sentences.
- Each character in the array occupies one byte of memory, and the last character must always be 0.
- The termination character ('\0') is important in a string since it is the only way to identify where the string ends.
- When we define a string as `char s[10]`, the character `s[10]` is implicitly initialized with the null in the memory.

Strings

- There are two ways to declare a string in C++ language.
 - By char array
 - By string literal
- Let's see the example of declaring string by char array in C++.
 - `char ch[10]={'t', 'u', 's', 'h', 'a', 'r', '\0'};`

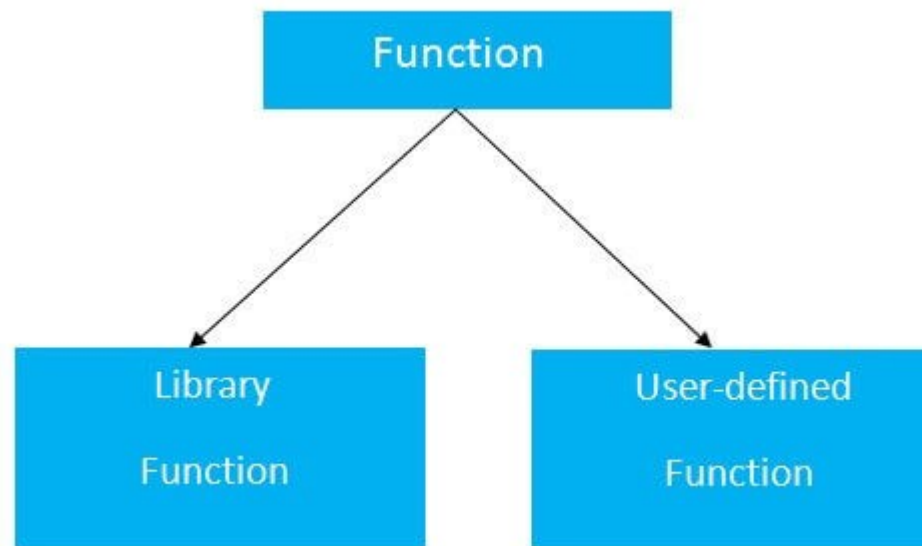
String Functions

| No. | Function | Description |
|-----|--|--|
| 1) | <code>strlen(string_name)</code> | returns the length of string name. |
| 2) | <code>strcpy(destination, source)</code> | copies the contents of source string to destination string. |
| 3) | <code>strcat(first_string, second_string)</code> | concatenates or joins first string with second string. The result of the string is stored in first string. |
| 4) | <code>strcmp(first_string, second_string)</code> | compares the first string with second string. If both strings are same, it returns 0. |
| 5) | <code>strrev(string)</code> | returns reverse string. |
| 6) | <code>strlwr(string)</code> | returns string characters in lowercase. |
| 7) | <code>strupr(string)</code> | returns string characters in uppercase. |

Functions

- In c, we can divide a large program into the basic building blocks known as function.
- The function contains the set of programming statements enclosed by {}. A function can be called multiple times to provide reusability and modularity to the C++ program.
- In other words, we can say that the collection of functions creates a program.
- The function is also known as procedure or subroutine in other programming languages.

Functions



Functions

- **Function declaration** A function must be declared globally in a c program to tell the compiler about the function name, function parameters, and return type.
- **Function call** Function can be called from anywhere in the program. The parameter list must not differ in function calling and function declaration. We must pass the same number of functions as it is declared in the function declaration.
- **Function definition** It contains the actual statements which are to be executed. It is the most important aspect to which the control comes when the function is called. Here, we must notice that only one value can be returned from the function.

Functions

- Function declaration
 - `return_type function_name (argument list);`
- Function call
 - `function_name (argument_list)`
- Function definition
 - `return_type function_name (argument list)`
`{function body;}`

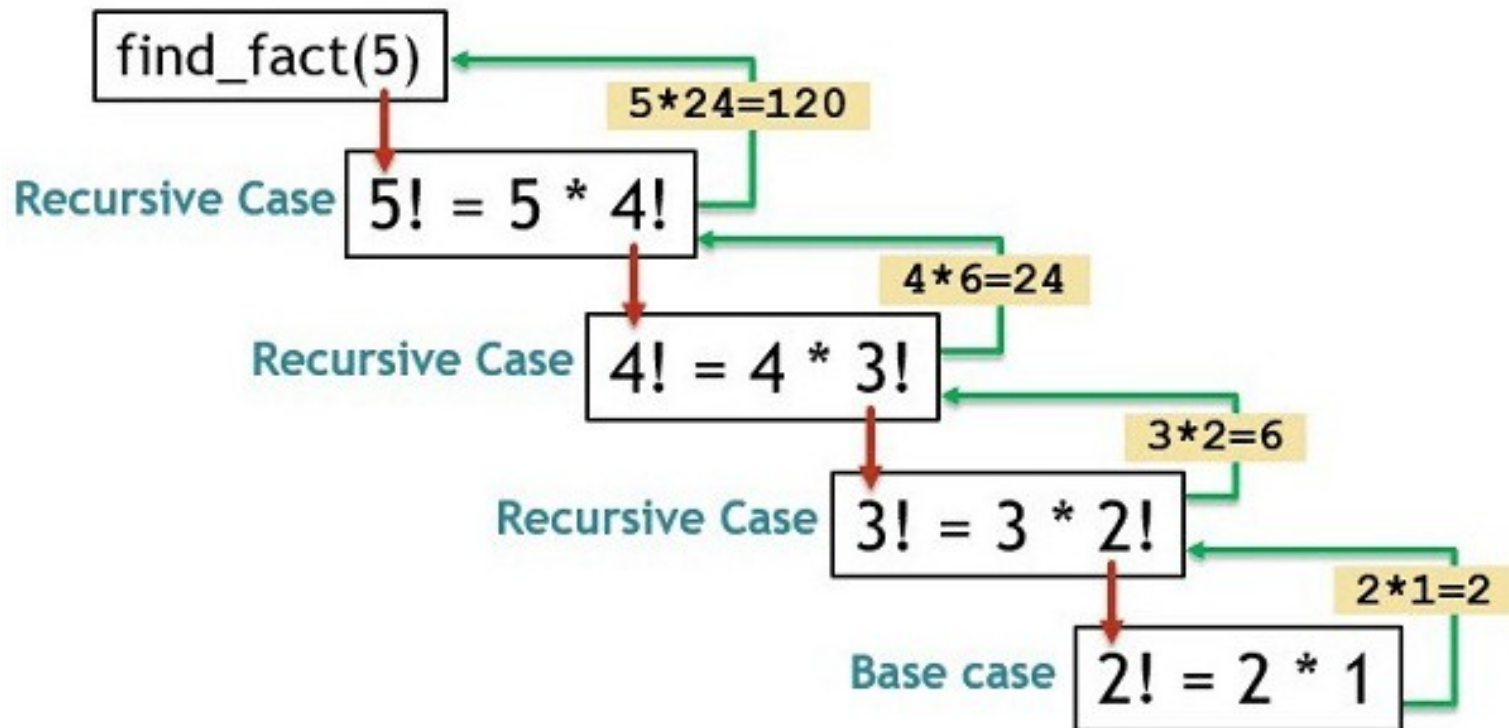
Recursion

- Recursion is the process which comes into existence when a function calls a copy of itself to work on a smaller problem.
- Any function which calls itself is called recursive function, and such function calls are called recursive calls.
- Recursion involves several numbers of recursive calls. However, it is important to impose a termination condition of recursion.
- Recursion code is shorter than iterative code however it is difficult to understand.

Recursion

- Recursion cannot be applied to all the problem, but it is more useful for the tasks that can be defined in terms of similar subtasks.
- For Example, recursion may be applied to sorting, searching, and traversal problems.
- Generally, iterative solutions are more efficient than recursion since function call is always overhead.
- Any problem that can be solved recursively, can also be solved iteratively. However, some problems are best suited to be solved by the recursion, for example, tower of Hanoi, Fibonacci series, factorial finding, etc.

Recursion

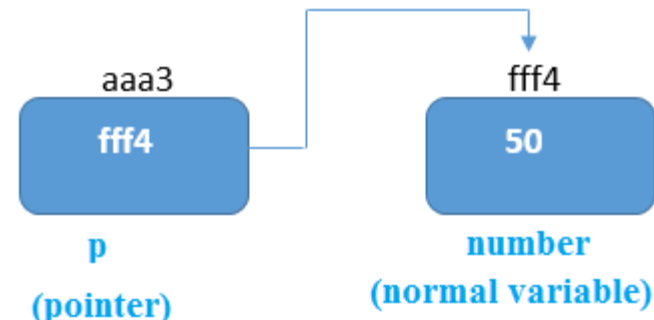


Pointers

- The pointer in C++ language is a variable which stores the address of another variable.
- This variable can be of type int, char, array, function, or any other pointer.
- The size of the pointer depends on the architecture. However, in 32-bit architecture the size of a pointer is 2 byte.
- Consider the following example to define a pointer which stores the address of an integer.
 - `int n = 10;`
 - `int* p = &n; // Variable p of type pointer is pointing to the address of the variable n of type integer.`

Pointers

- The pointer in C++ can be declared using * (asterisk symbol).
- It is also known as indirection pointer used to dereference a pointer.
 - `int *a;`//pointer to int
 - `char *c;`//pointer to char



Thank you

This presentation is created using LibreOffice Impress 7.4.1.2, can be used freely as per GNU General Public License



@mitu_skillologies



@mITuSkillologies



@mitu_group



@mitu-skillologies



@MITUSkillologies

kaggle

@mituskillologies

Web Resources

<https://mitu.co.in>

<http://tusharkute.com>



@mituskillologies

contact@mitu.co.in
tushar@tusharkute.com