

[Suggest a Topic](#)[Login](#)[Write an Article](#)

C++ bitset interesting facts

Bitset is a container in C++ Standard Template Library for dealing with data at the bit level.

1. A bitset stores bits (elements with only two possible values: 0 or 1). We can however get the part of a string by providing positions to bitset constructor (Positions are with respect to string position from left to right)

Example:

```
// C++ program to demonstrate that we can get part of a
// bit string in bitset.
#include <bitset>
#include <string>
#include <iostream>

int main()
{
    std::string bit_string = "110010";
    std::bitset<8> b1(bit_string);           // [0, 0, 1, 1, 0, 0, 1, 0]

    // string from position 2 till end
    std::bitset<8> b2(bit_string, 2);        // [0, 0, 0, 0, 0, 0, 1, 0]

    // string from position 2 till next 3 positions
    std::bitset<8> b3(bit_string, 2, 3);    // [0, 0, 0, 0, 0, 0, 0, 1]

    std::cout << b1 << '\n' << b2 << '\n' << b3 << '\n';

    return 0;
}
```

[Run on IDE](#)

Output:

```
00110010
00000010
00000001
```

2. We can construct a bitset using the characters in the `std::basic_string _str`. An optional starting position `_pos` and length `_n` can be provided, as well as characters denoting alternate values for set (`_one`) and

unset (`_zero`) bits.

Syntax:

```
std::bitset b1(str, pos, n, zero, one);
str      : string used to initialize the bitset
pos      : a starting offset into str
n        : number of characters to use from str
zero     : alternate character for unset bits in str
one      : alternate characters for set bits in str
```

- If `_pos > str.size()`, this constructor throws `std::out_of_range`.
- If any characters examined in `_str` is not zero or one, it throws `std::invalid_argument`.

```
// C++ program to demonstrate that we can construct bitset using
// alternate characters for set and unset bits.
#include <bitset>
#include <string>
#include <iostream>

int main()
{
    // string constructor using custom zero/one digits
    std::string alpha_bit_string = "aBaaBBaB";
    std::bitset<8> b1(alpha_bit_string, 0, alpha_bit_string.size(),
                     'a', 'B');           // [0,1,0,0,1,1,0,1]

    std::cout << b1 << '\n';
}
```

[Run on IDE](#)

Output:

```
01001101
```

3. Constructs an object of class `bitset`, initializing the `N` bits to values that correspond to the characters provided in a c-style character string of zeros and ones. You call the constructor without casting the string into a string type. It also has two optional parameters, `_Zero` and `_One`, which indicate what character in `_Str` is to be interpreted to mean a 0 bit and a 1 bit, respectively.

```
#include <bitset>
#include <iostream>

int main()
{
    // char* constructor using custom digits
    std::bitset<8> b1("XXXYYYY", 8, 'X', 'Y'); // [0, 0, 0, 0, 1, 1, 1, 1]
    std::cout << b1 << '\n';
}
```

[Run on IDE](#)

Output:

```
00001111
```

Bitset Operations

1. `std::bitset::to_string()`

Converts the contents of the bitset to a string. Uses zero to represent bits with value of false and one to represent bits with value of true. The resulting string contains N characters with the first character corresponds to the last (N-1th) bit and the last character corresponding to the first bit. Also, we can pass the characters used to print true and false value through the parameters.

Example:

```
// C++ program to demonstrate that we can convert contents
// of bitset to a string.
#include <iostream>
#include <bitset>

int main()
{
    std::bitset<8> b(42);
    std::cout << b.to_string() << '\n'
              << b.to_string('*') << '\n'
              << b.to_string('0', 'X') << '\n';
}
```

[Run on IDE](#)

Output:

```
00101010
**1*1*1*
00X0X0X0
```

2. `std::bitset::to_ulong()`:

Converts the contents of the bitset to an unsigned long integer. The first bit of the bitset corresponds to the least significant digit of the number and the last bit corresponds to the most significant digit. Function throws `std::overflow_error` if the value cannot be represented in unsigned long.

Example:

```
// C++ program to demonstrate that we can get value of bitset
// as unsigned long integer.
#include <iostream>
#include <bitset>

int main()
{
    std::bitset<5> b(5);
    std::cout << b.to_ulong() << '\n';
}
```

[Run on IDE](#)

Output:

5

This article is contributed by **Rohit Kasle**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Practice Tags : Bit Magic STL C CPP

Article Tags : Bit Magic C C++ CPP-bitset STL



Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

[Login to Improve this Article](#)

Recommended Posts:

[C++ bitset and its application](#)

[Find profession in a special family](#)

[Bits manipulation \(Important tactics\)](#)

[Builtin functions of GCC compiler](#)

[Swap two nibbles in a byte](#)

[Remove duplicates from a string in O\(1\) extra space](#)

[Calculate the total fine to be collected](#)

[Sum of bitwise OR of all subarrays](#)

[Value in a given range with maximum XOR](#)

[Check if a number can be expressed as \$2^x + 2^y\$](#)

[\(Login to Rate\)](#)**3.6**Average Difficulty : **3.6/5.0**
Based on **5** vote(s)[Add to TODO List](#)[Mark as DONE](#)[Feedback](#)[Add Notes](#)[Improve Article](#)[Basic](#)[Easy](#)[Medium](#)[Hard](#)[Expert](#)

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)[Share this post!](#)

A computer science portal for geeks

710-B, Advant Navis Business Park,
Sector-142, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

[About Us](#)
[Careers](#)
[Privacy Policy](#)
[Contact Us](#)

PRACTICE

[Company-wise](#)
[Topic-wise](#)
[Contests](#)
[Subjective Questions](#)

LEARN

[Algorithms](#)
[Data Structures](#)
[Languages](#)
[CS Subjects](#)
[Video Tutorials](#)

CONTRIBUTE

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)

@geeksforgeeks, Some rights reserved

