Literals in C

In C, **literals** are the constant values that are assigned to the variables. They represent fixed values whose meaning does not change during program execution. **For example**, "float x = 3.14" is an expression and the value 3.14 is referred to as an integer literal.

There are 4 types of literals in C corresponding to the different data types in C:

1. Integer Literals

Integer literals are used to represent and store the integer values only. It can have prefix and suffix that indicates different properties of the literal

Integer Literals with Prefix

The prefix of the integer literal indicates the base on which it is to be read.

- **No Prefix:** Decimal value (e.g., 20)
- **0x or 0X:** Hexadecimal value (e.g., 0x1A)
- **0:** Octal value (e.g., 016)
- **0b or 0B:** Binary value (GCC extension, not part of the standard, e.g., 0b11)

Example:

C

```c
#include <stdio.h>
int main()
{
    int a = 20; // 20
    int b = 0x1A; // 26
    int c = 016; // 14
    int d = 0b11; // 3
    printf("%d %d %d %d", a, b, c, d);
    return 0;
}
```

Output

```
20 26 14 3
```

Integer Literals with Suffix

The suffixes of the integer literal indicate the type in which it is to be read.

- **u or U**: Unsigned (e.g., 124U)
- **l or L**: Long (e.g., 124L)
- **ll or LL**: Long Long (e.g., 124LL)

Example:

C

```c
#include <stdio.h>
int main()
{
    int a = 124;
    unsigned int b = 124u;
    long int c = 124L;
    long long d = 124LL;
    return 0;
}
```

2. Floating-Point Literals

Floating-point literals are used to represent real numbers. The real number has an integer part and a fractional part. In floating-point literals, allowed prefix is **"0x"** or allowed suffix are **"f for float and l for long double".**

Example:

C

```c
#include <stdio.h>
int main()
{
    float a = 10.5f;
    double b = 10.515;
    long double c = 10.515l;
    float d = 2.1e4f; // 2.1x1^4 = 21000
    double e = 200.1e-80; // 200.1x10^-80 = 2.001x10^-98
    double f =  0x1A.1p2; // 26.0625x2^2 = 104.25
    printf("%g %g %Lg %g %g %g", a, b, c, d, e, f);
    return 0;
}
```

Output

```
10.5 10.515 10.515 21000 2.001e-78 104.25
```

By default in C, if you do not write any suffix with floating it treated as double.

3. & 4. Character and String Literals

Character literals refer to the literal that is used to store a single character within a single quote.

String literals are similar to that character literals, except that they can store multiple characters and uses a double quote to store the same. It can also accommodate the special characters. String literals are immutable and stored in a read-only memory area.

Example:

C

```
#include <stdio.h>
int main()
{
    char c = 'g';
    const char *s = "gfg";
    printf("%c %s", c, s);
    return 0;

```