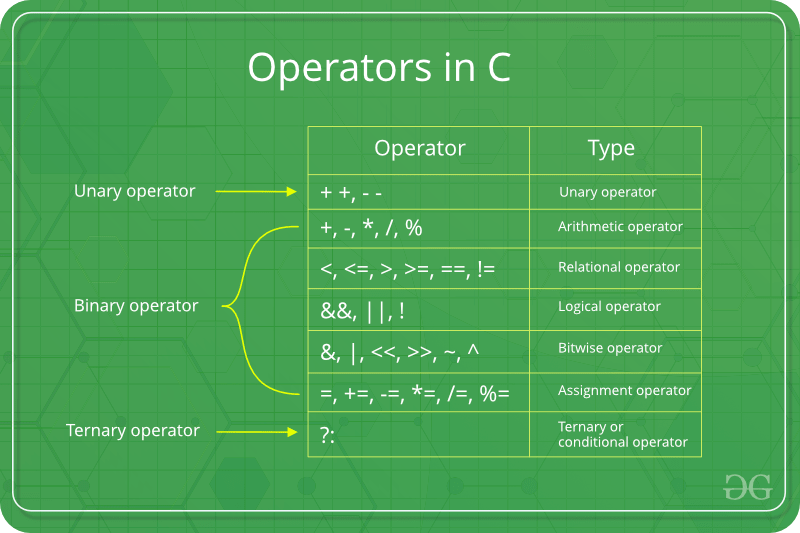
Operstors



* Given a set of numbers where all elements occur even number of times except one number, find the odd occurring number

#include<stdio.h>

**int** func(**int** arr[], **int** n)

{

**int** res = 0, i;

**for** (i = 0; i < n; i++)

res ^= arr[i];

**return** res;

}

**int** main(**void**)

{

**int** arr[] = {1,2,3,4,1,2,3,4,9};

**int** n = **sizeof**(arr) / **sizeof**(arr[0]);

printf("The odd occurring element is %d ",func(arr, n));

**return** 0;

}

* Swap two numbers without using temporary variable

Method 1 x = x + y; // x now becomes 15

     y = x - y; // y becomes 10

     x = x - y;

method 2

x = x \* y;

    y = x / y;

    x = x / y;

method 3

x = x ^ y;

    y = x ^ y;

    x = x ^ y;

* Add two numbers without using arithmatic operators

#include<stdio.h>

**int** Add(**int** x, **int** y)

{

**while** (y != 0)

{

**int** carry = x & y;

x = x ^ y;

y = carry << 1;

}

**return** x;

}

**int** main()

{

printf("%d", Add(15, 32));

**return** 0;

}

* **The left-shift and right-shift operators are equivalent to multiplication and division by 2 respectively.**
* **The & operator can be used to quickly check if a number is odd or even.**

int x = 19;

    (x & 1) ? printf("Odd") : printf("Even");

* Sizeof() operator is used to find the the number of arrayr.
* sizeof is greatly used in dynamic memory allocation.

int\* ptr = (int\*)malloc(10 \* sizeof(int));

* postfix>>prefix>>deference
* stringize operator # - turns the argument it precedes into a quoted string.

#include <stdio.h>

#define mkstr(s) #s //it will concert the following to string

**int** main(**void**)

{

printf(mkstr(imadi\n)); // it is readed as printf(("imadi\n");

**return** 0;

}

* # is a pre-processor directive.
* the ‘#define’ directive is used to define a macro.
* #define LIMIT 5 – til will set value of limit = 5 globally.
* ## is known as token pasting operator -

#include <stdio.h>

#define concat(a, b) a##b

int main(void)

{

    int xy = 30;

    printf("%d", concat(x, y));

    return 0;

}