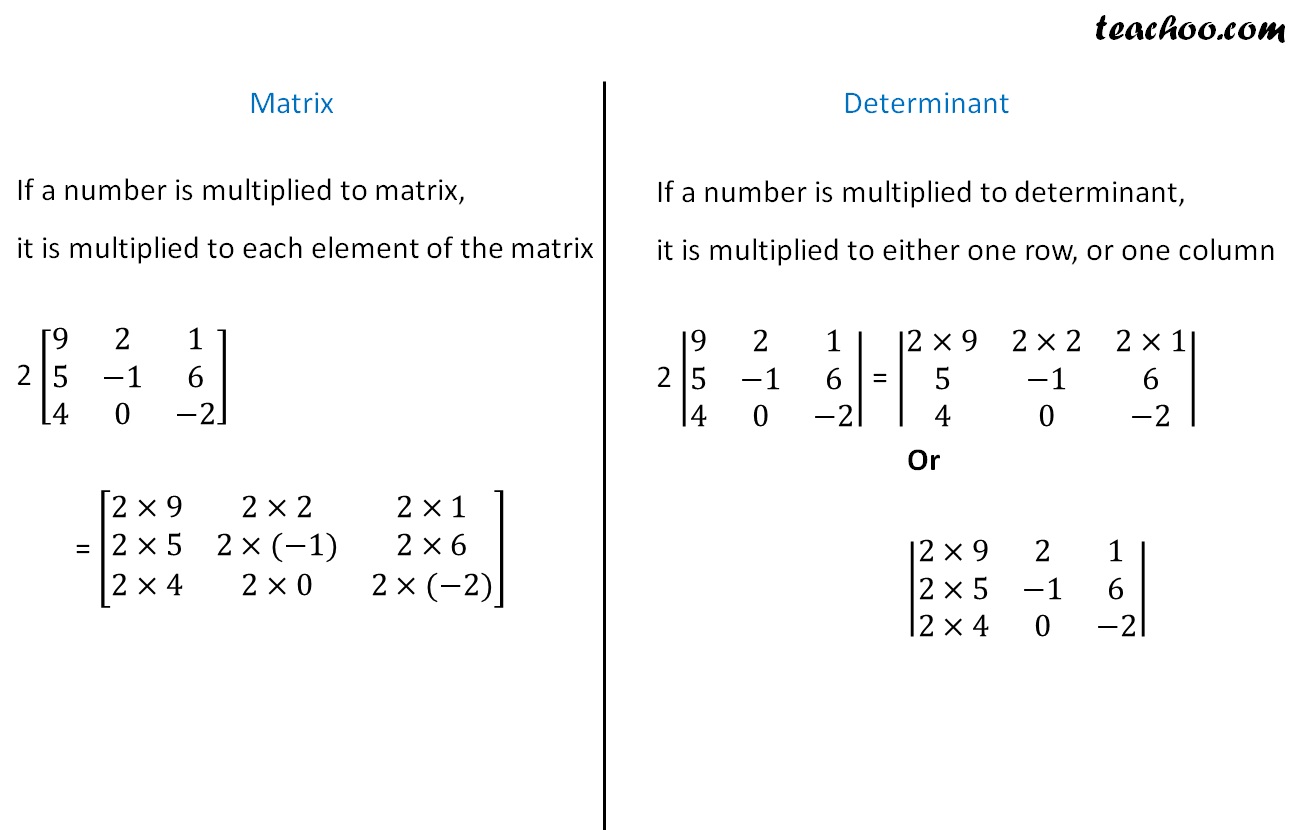
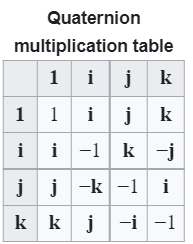
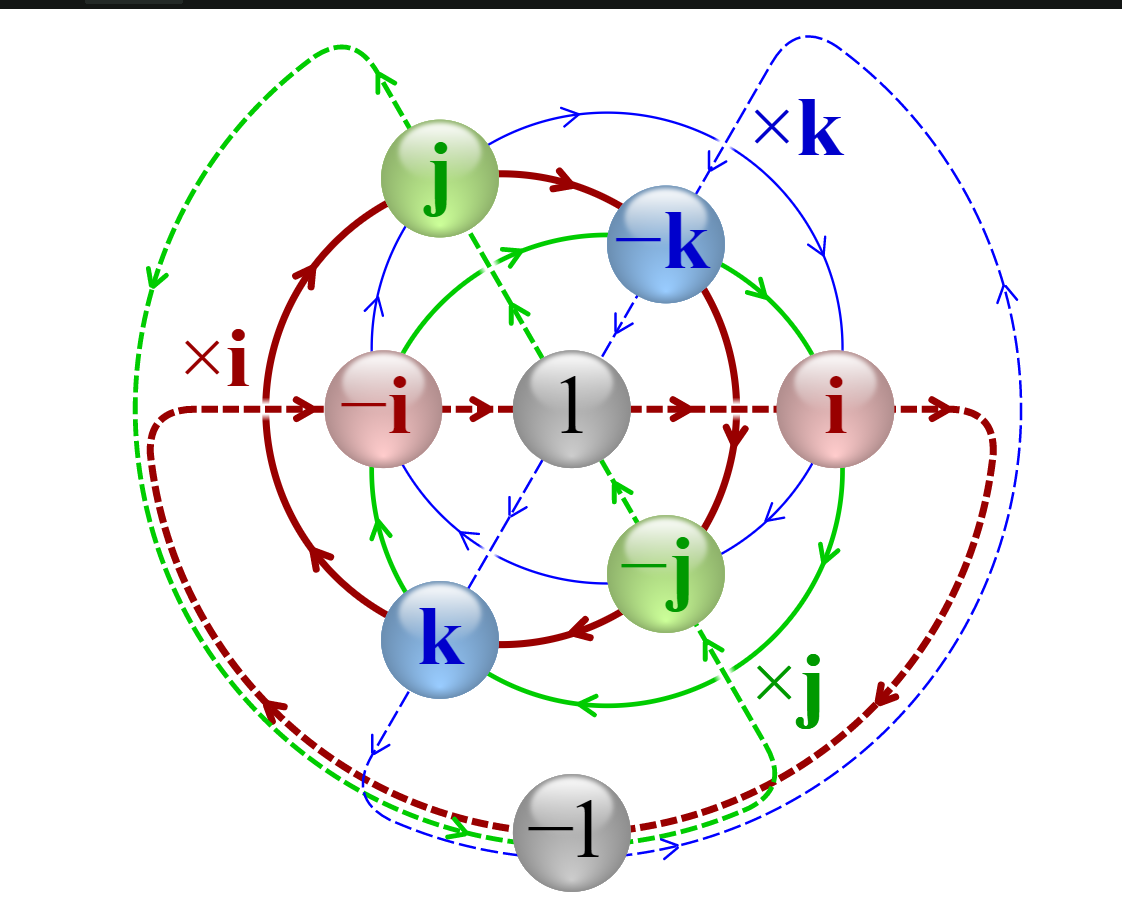
**PROGRAMMING MATH & FUNCTIONS**

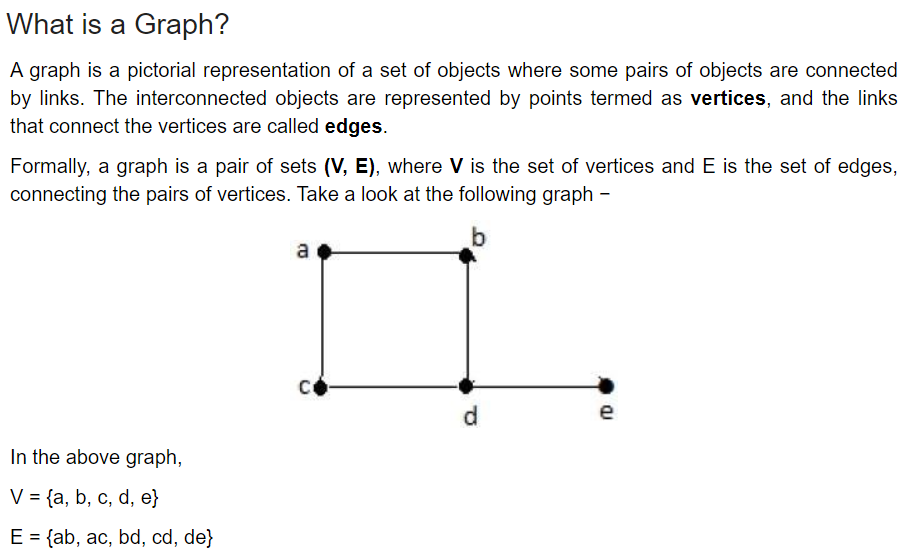
**1)Matrix** (матрица, детерминант(определитель))

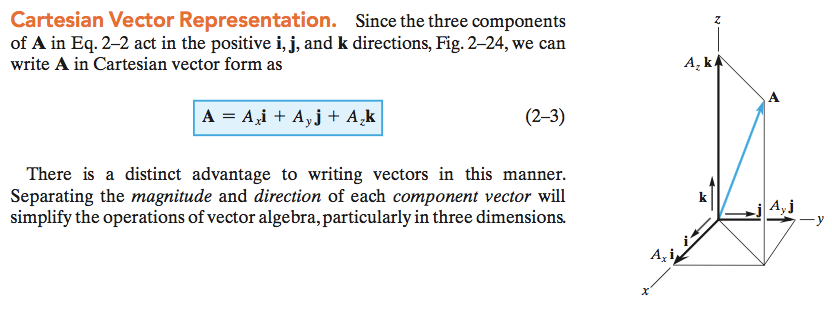
**3)Quaternion**

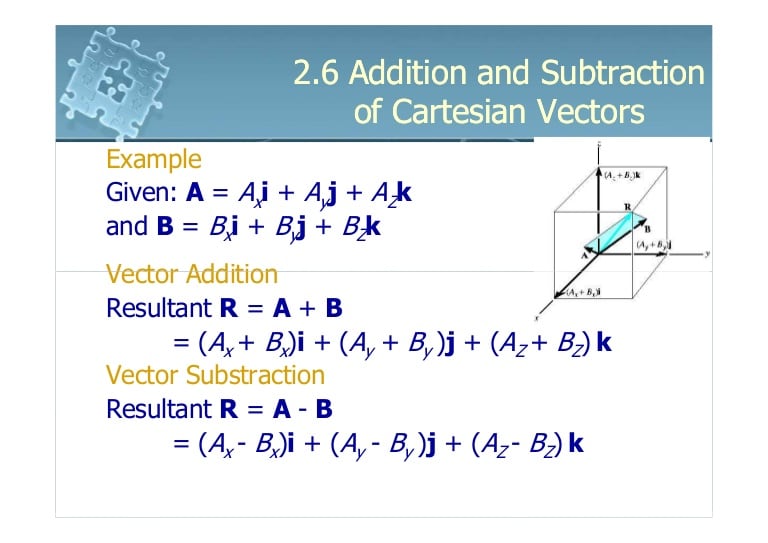
In [mathematics](https://en.wikipedia.org/wiki/Mathematics), the **quaternion** [number system](https://en.wikipedia.org/wiki/Number_system) extends the [complex numbers](https://en.wikipedia.org/wiki/Complex_number). Quaternions were first described by Irish mathematician [William Rowan Hamilton](https://en.wikipedia.org/wiki/William_Rowan_Hamilton) in 1843[[1]](https://en.wikipedia.org/wiki/Quaternion#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Quaternion#cite_note-2) and applied to [mechanics](https://en.wikipedia.org/wiki/Mechanics) in [three-dimensional space](https://en.wikipedia.org/wiki/Three-dimensional_space). Hamilton defined a quaternion as the [quotient](https://en.wikipedia.org/wiki/Quotient) of two [*directed lines*](https://en.wikipedia.org/wiki/Directed_line_segment) in a three-dimensional space,[[3]](https://en.wikipedia.org/wiki/Quaternion#cite_note-3) or, equivalently, as the quotient of two [vectors](https://en.wikipedia.org/wiki/Vector_(geometry)).[[4]](https://en.wikipedia.org/wiki/Quaternion#cite_note-4) Multiplication of quaternions is [noncommutative](https://en.wikipedia.org/wiki/Noncommutative).



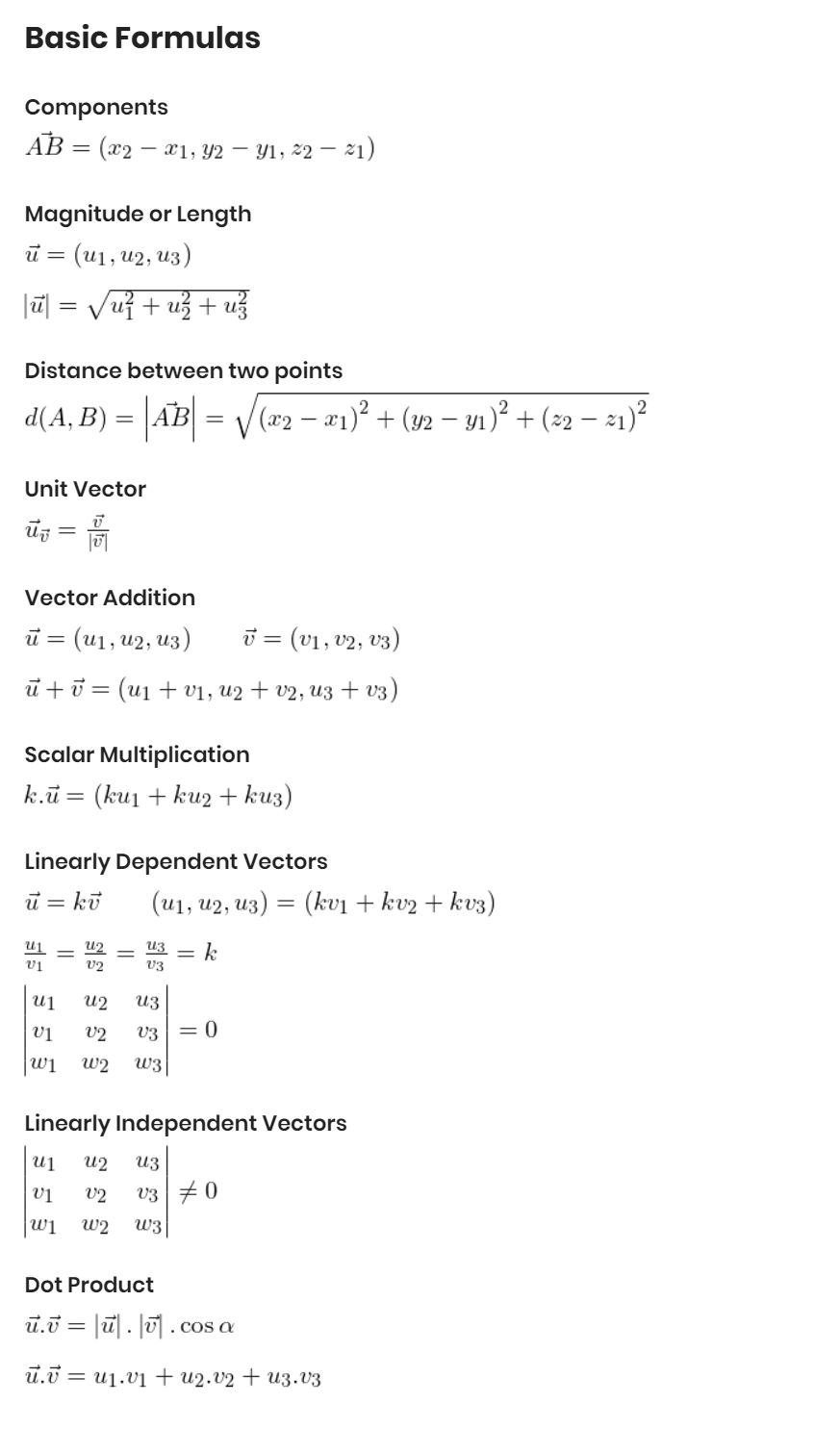


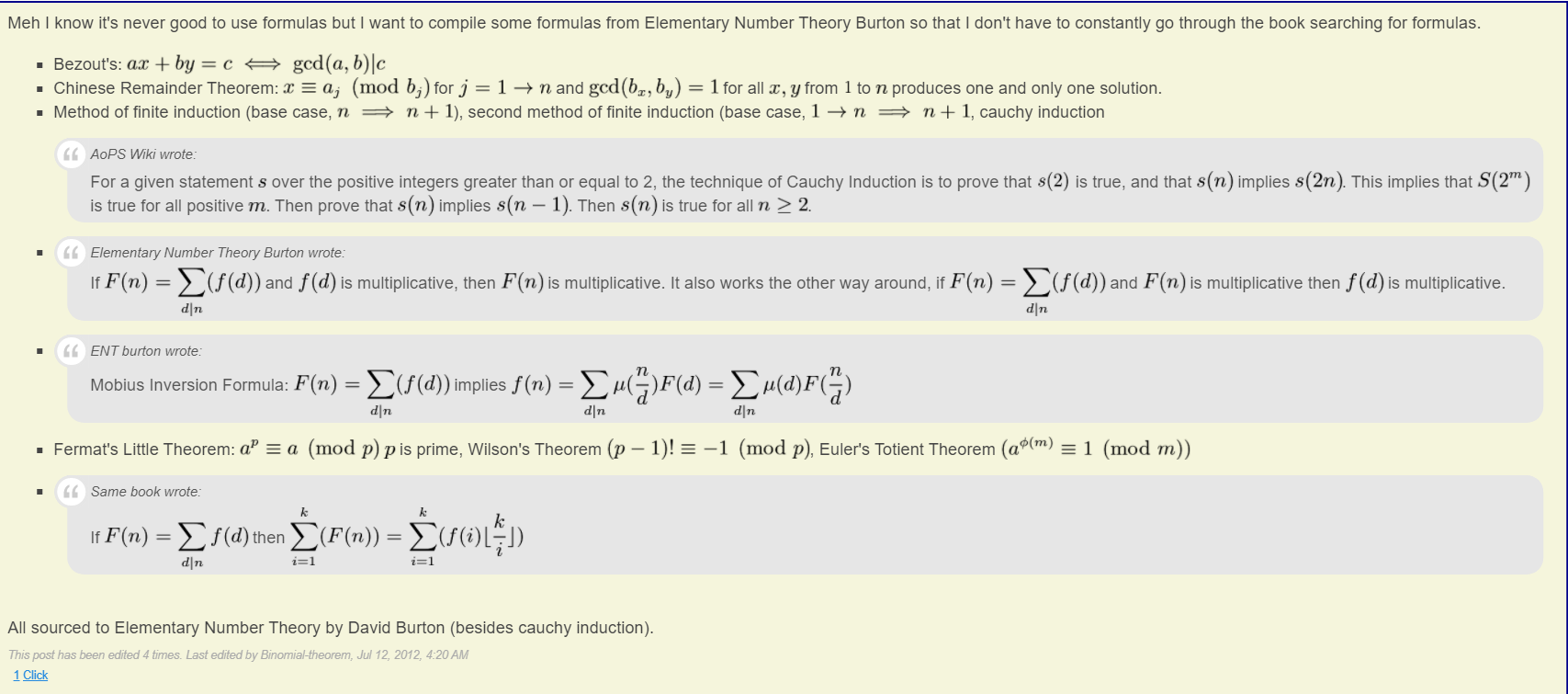
**4) Graph Theory** (More here: https://www.tutorialspoint.com/graph\_theory/graph\_theory\_quick\_guide.htm)

**5) Cartesian Vector**



**6) Vectors**

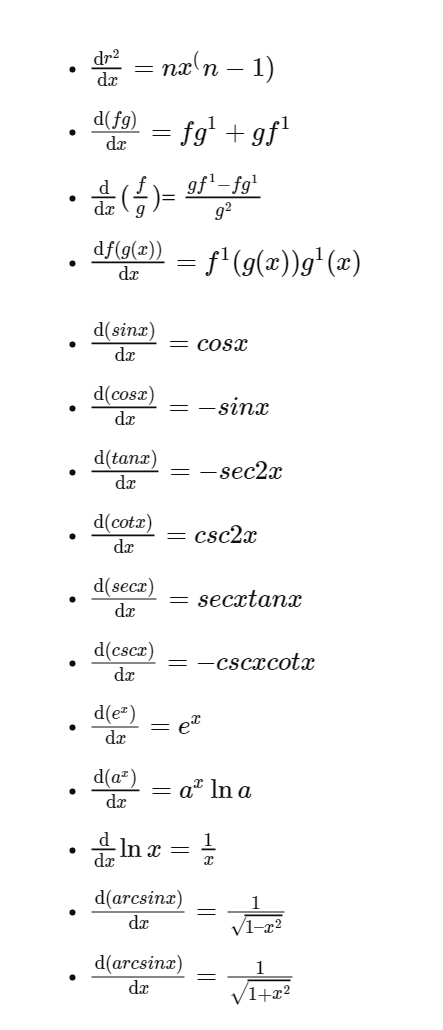
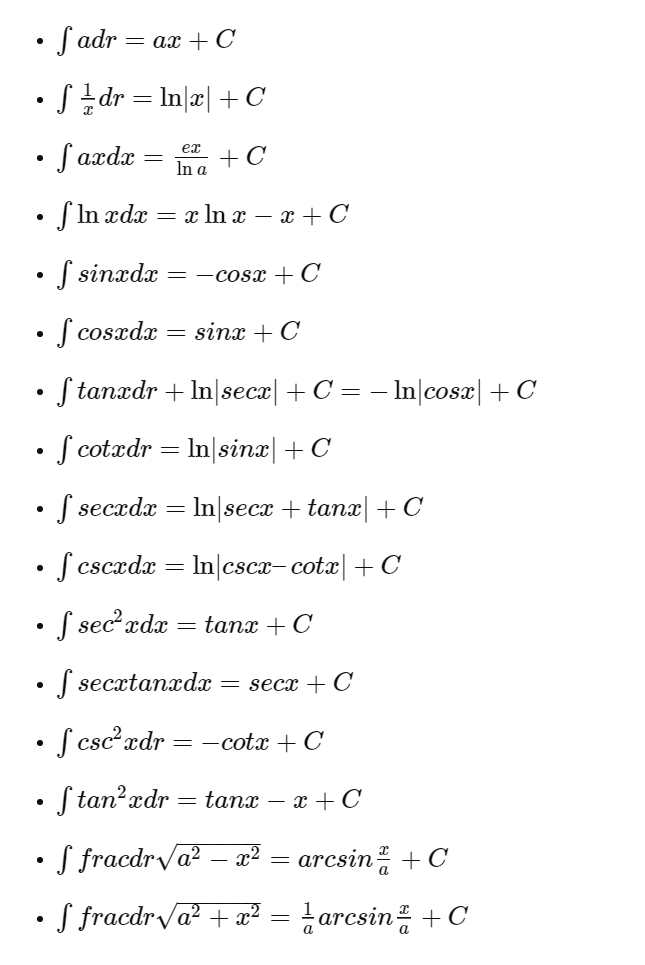


**7) Number Theory**

**8)Calculus theory**

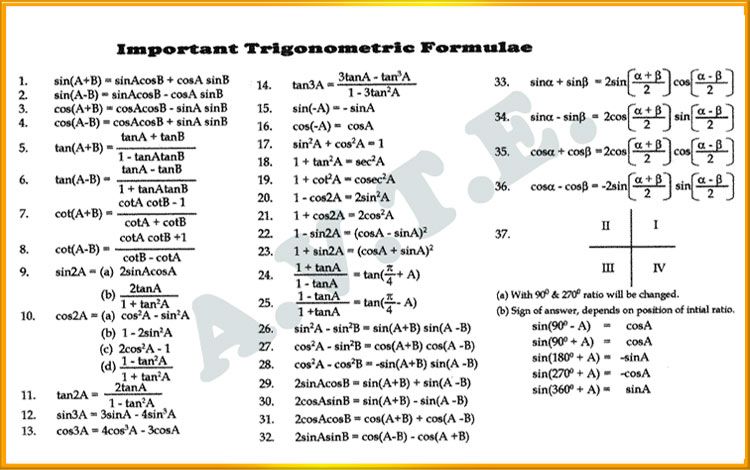
Calculus is the branch of mathematics, which deals in the study rate of change and its application in solving the equations. Differential Calculus deals with the rates of change and slopes of curves. Integral Calculus deals mainly with the accumulation of quantities and the areas under and between curves. In calculus, we use the fundamental notions of convergence of infinite sequences and infinite series to a well-defined limit. its value is less than any positive [real number](https://www.toppr.com/guides/maths/real-numbers/). According to this point of view, we can say that calculus is a collection of techniques for manipulating infinitesimals. In calculus the symbols dxdxanddydy were taken to be infinitesimal, and the derivative dydxdy/dx was simply their ratio.

**Differential Calculus Integral Calculus**



**9)Trigonometry Formulas**

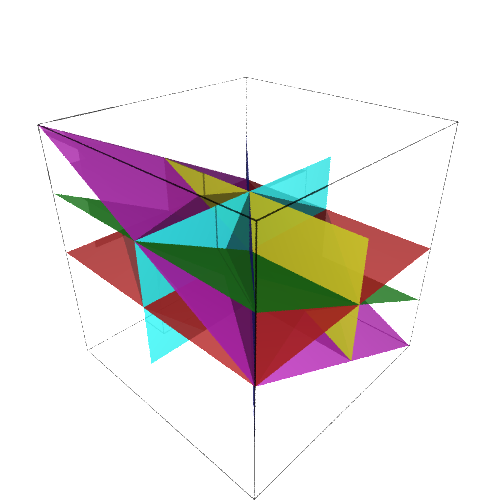
**Trigonometry** (from [Greek](https://en.wikipedia.org/wiki/Ancient_Greek) *[trigōnon](https://en.wiktionary.org/wiki/%CF%84%CF%81%CE%AF%CE%B3%CF%89%CE%BD%CE%BF%CE%BD" \o "wikt:τρίγωνον)*, "triangle" and *[metron](https://en.wiktionary.org/wiki/%CE%BC%CE%AD%CF%84%CF%81%CE%BF%CE%BD" \o "wikt:μέτρον)*, "measure"[[1]](https://en.wikipedia.org/wiki/Trigonometry#cite_note-1)) is a branch of [mathematics](https://en.wikipedia.org/wiki/Mathematics) that studies relationships between side lengths and [angles](https://en.wikipedia.org/wiki/Angle) of [triangles](https://en.wikipedia.org/wiki/Triangle)



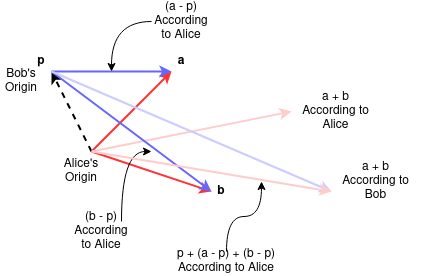
**10)Geometry**

In [geometry](https://en.wikipedia.org/wiki/Geometry), a **hyperplane** is a subspace whose [dimension](https://en.wikipedia.org/wiki/Dimension) is one less than that of its [ambient space](https://en.wikipedia.org/wiki/Ambient_space). If a space is 3-dimensional then its hyperplanes are the 2-dimensional [planes](https://en.wikipedia.org/wiki/Plane_(geometry)), while if the space is 2-dimensional, its hyperplanes are the 1-dimensional [lines](https://en.wikipedia.org/wiki/Line_(geometry)). This notion can be used in any general [space](https://en.wikipedia.org/wiki/Space_(mathematics)) in which the concept of the dimension of a [subspace](https://en.wikipedia.org/wiki/Topological_subspace) is defined.

• A hyperplane is a higher-dimensional generalization of lines and planes. The equation of a hyperplane is w · x + b = 0, where w is a vector normal to the hyperplane and b is an offset. Note that we can multiply by any constant and preserve the equality; if we multiply by 1/kwk, we get a new equation ˆw · x + b 0 = 0, where ˆw = w/kwk is the unit normal vector and b 0 = b/kwk is the distance from the hyperplane to the origin.



In an **affine space**, there is no distinguished point that serves as an origin. Hence, no vector has a fixed origin and no vector can be uniquely associated to a point. In an affine space, there are instead [*displacement vectors*](https://en.wikipedia.org/wiki/Displacement_vector), also called [*translation*](https://en.wikipedia.org/wiki/Translation_(geometry)) vectors or simply *translations*, between two points of the space.[[1]](https://en.wikipedia.org/wiki/Affine_space#cite_note-1) Thus it makes sense to subtract two points of the space, giving a translation vector, but it does not make sense to add two points of the space. Likewise, it makes sense to add a displacement vector to a point of an affine space, resulting in a new point translated from the starting point by that vector.



More: <https://en.wikipedia.org/wiki/Affine_space>

**11) Logic Arithmetic**

Uses logical operators in programming language, for every language they different (&&,||,==)

**12) Boolean Arithmetic**

(True or false)

**13) Bit Manipulation**

(And, Or, Xor, Not, Xnor, Neg,Nand, Nor, Left Shift, Right Shift)

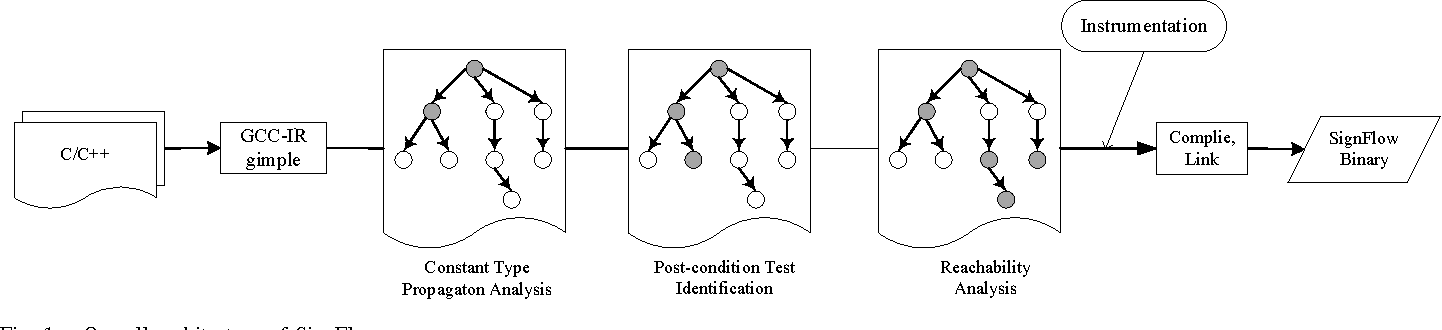
Bit manipulation is the **act of algorithmically manipulating bits or other pieces of data shorter than a word**. Computer programming tasks that require bit manipulation include low-level device control, error detection and correction algorithms, data compression, encryption algorithms, and optimization.

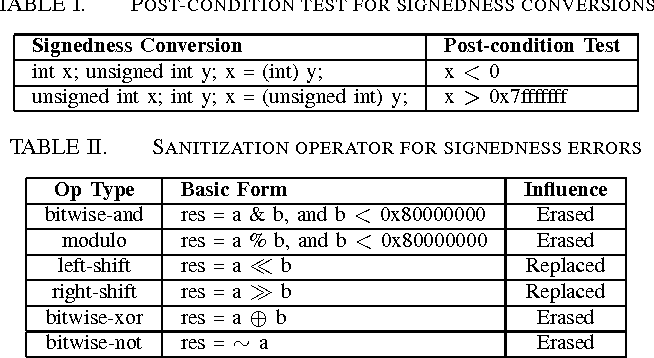
More on: <https://www.hackerearth.com/practice/basic-programming/bit-manipulation/basics-of-bit-manipulation/tutorial/>

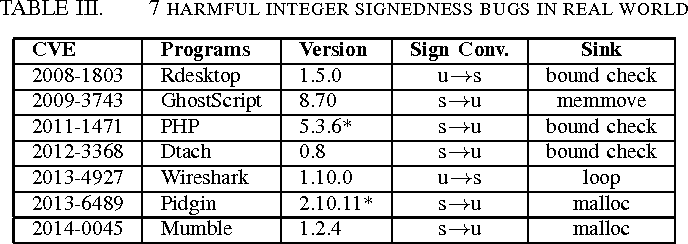
**14)Comparing Numbers**

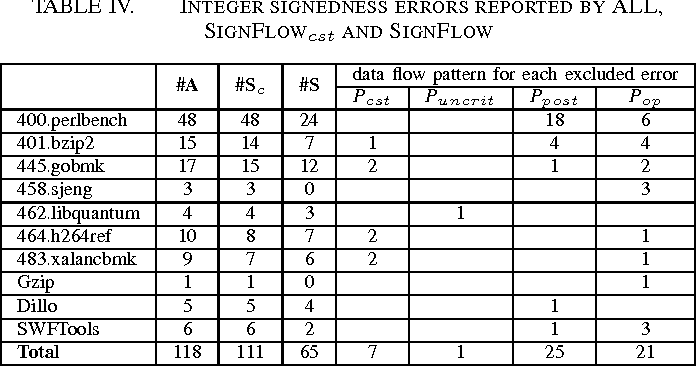
**15)Signedness**

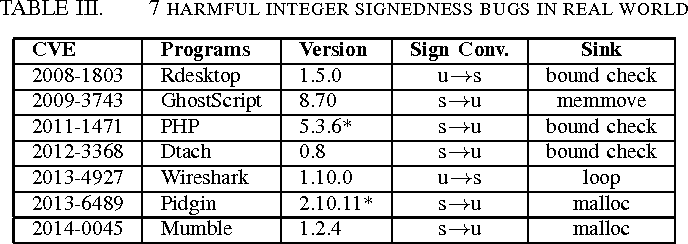
In computing, signedness is **a property of data types representing numbers in computer programs**. A numeric variable is signed if it can represent both positive and negative numbers, and unsigned if it can only represent non-negative numbers (zero or positive numbers).

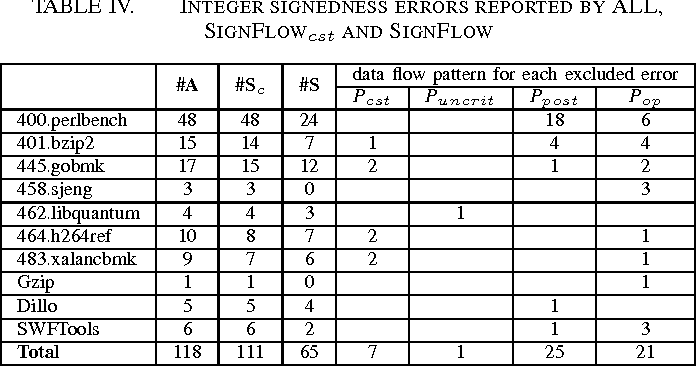












**16) Add, Subtract, Multiply, Division**

**17) Tables**

**LogTables**

Every SQL Server database has a **transaction log that records all transactions and the database modifications made by each transaction**. The transaction log is a critical component of the database. If there is a system failure, you will need that log to bring your database back to a consistent state

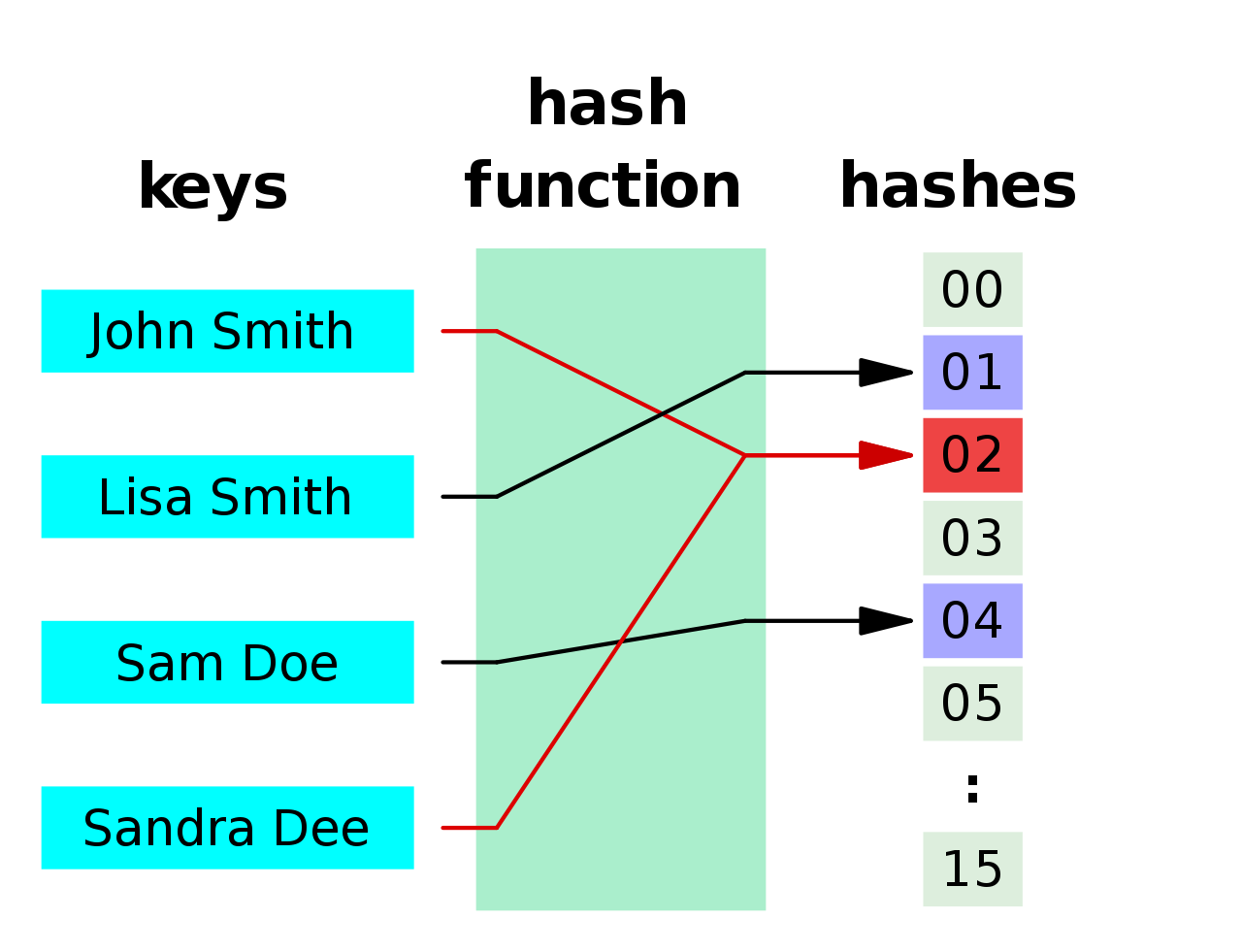
Table logging is **a track mechanism**, that makes possible to record user activity in a system It is useful if you want to know: Who made a change? ... As the parameter rec/client is used for controlling table logging at R/3 system level

**Division Tables**

Businesses often require reports that require more than the classic set operators. Surprisingly, a business requirement can often be expressed neatly in terms of the DIVISION relationship operator: How can this be done with SQL Server? Joe Celko opens up the 'Manga Guide to Databases', meets the Database Fairy, and is inspired to explain DIVISION.

**18) Hash Function**

A **hash function** is any [function](https://en.wikipedia.org/wiki/Function_(mathematics)) that can be used to map [data](https://en.wikipedia.org/wiki/Data_(computing)) of arbitrary size to fixed-size values. The values returned by a hash function are called *hash values*, *hash codes*, *digests*, or simply *hashes*. The values are usually used to index a fixed-size table called a [*hash table*](https://en.wikipedia.org/wiki/Hash_table). Use of a hash function to index a hash table is called *hashing* or *scatter storage addressing*.



**20) Modulo (%)**

In computing, the modulo operation **returns the remainder or signed remainder of a division**, after one number is divided by another (called the modulus of the operation). ... The modulo operation is to be distinguished from the symbol mod, which refers to the modulus (or divisor) one is operating from.

**21) Power/Exponent(pow)**

Return Type: The function returns the number base raised to the power. The type of this method is *System.Double*

Examples:

Input : base = 8, power =2

Output : 64

Input : base = 2.5, power =3

Output : 15.625

**22)Absolute numbers (abs)**

The **Math.abs()** function returns the absolute value of a number. That is, it returns x if x is positive or zero, and the negation of x if x is negative.

function difference(a, b) {

return Math.abs(a - b);

}

console.log(difference(3, 5));

// expected output: 2

console.log(difference(5, 3));

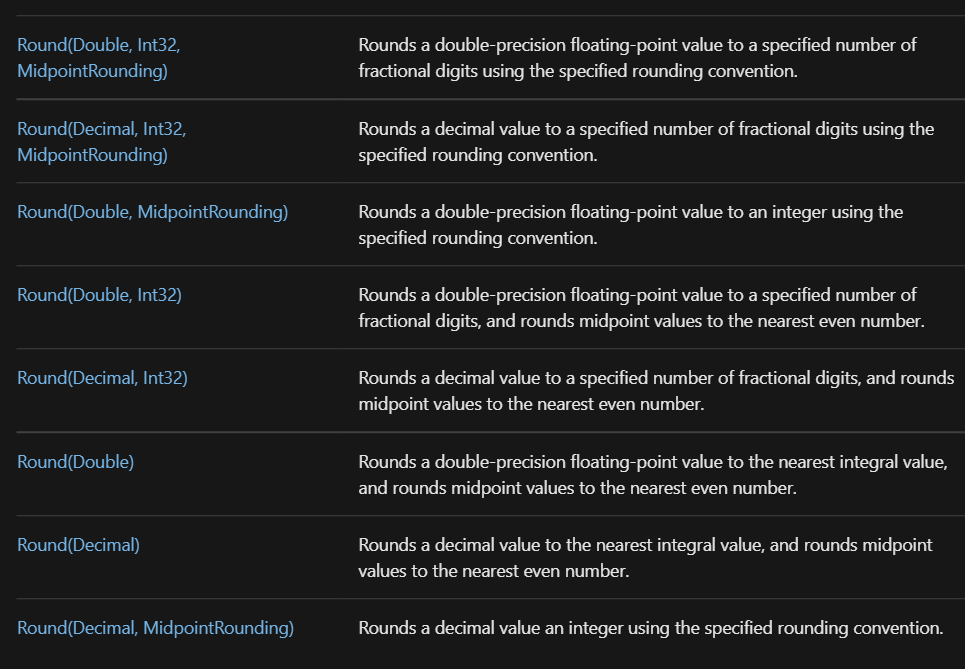
// expected output: 2

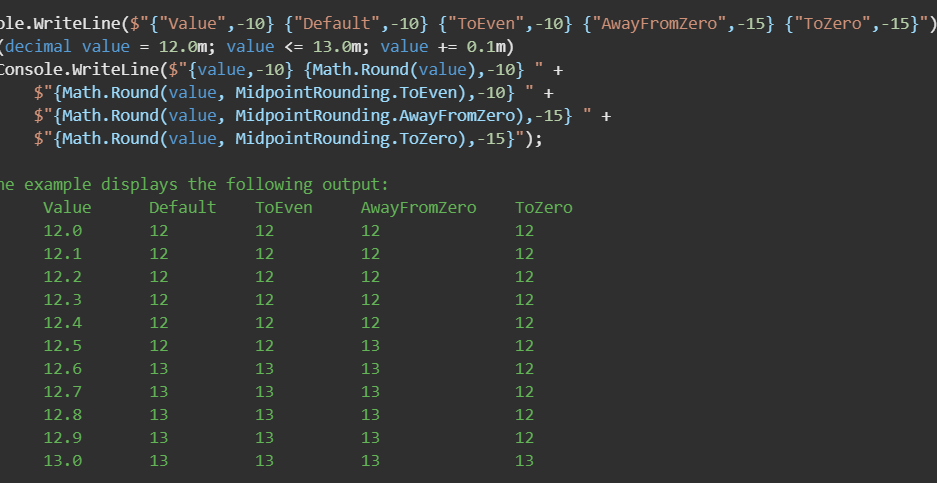
console.log(difference(1.23456, 7.89012));

// expected output: 6.6555599999999995

**23)Rounding off(roundf)**

Rounds a value to the nearest integer or to the specified number of fractional digits.





**24)Rounding up (ceil)**

Returns

The closest integer greater than or equal to *x*.

Description

Math.ceil( ) computes the ceiling function—i.e., it returns the closest integer value that is greater than or equal to the function argument. Math.ceil( ) differs from Math.round( ) in that it always rounds up, rather than rounding up or down to the closest integer. Also note that Math.ceil( ) does not round negative numbers to larger negative numbers; it rounds them up toward zero.

Example

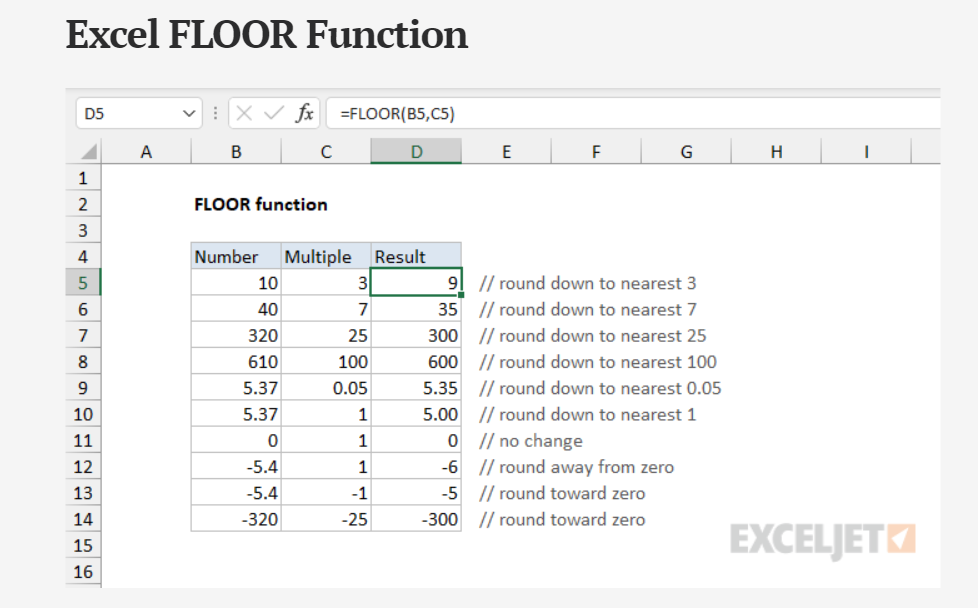
a = Math.ceil(1.99); // Result is 2.0

b = Math.ceil(1.01); // Result is 2.0

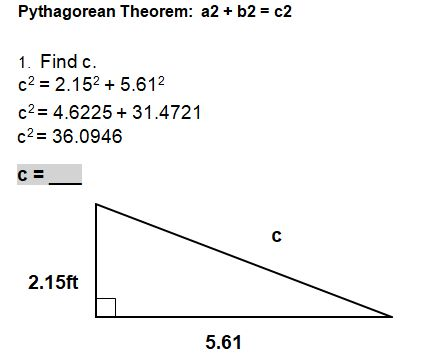
c = Math.ceil(1.0); // Result is 1.0

d = Math.ceil(-1.99); // Result is -1.0

**25)Rounding down (floor)**

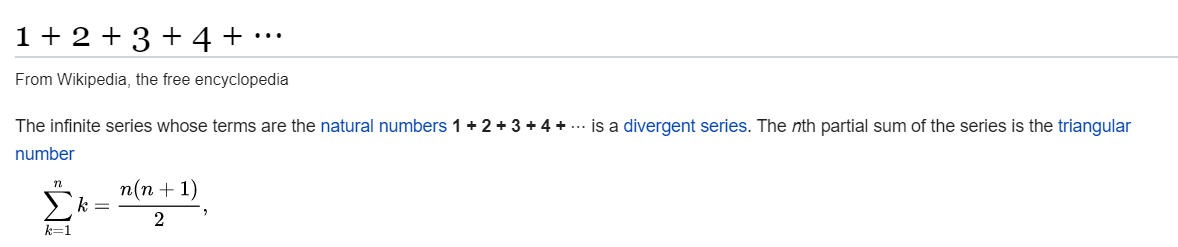


**26)Pyth.Thm. (a^2+b^2+c^2)**



**27)Floating Point Arithmetic (how floating points work)** [**https://en.wikipedia.org/wiki/IEEE\_754**](https://en.wikipedia.org/wiki/IEEE_754)

**28)Arithmetic sequence (1,2,3)**



// Java program to check if a given array

// can form arithmetic progression

import java.util.Arrays;

class GFG {

    // Returns true if a permutation of

    // arr[0..n-1] can form arithmetic

    // progression

    static boolean checkIsAP(int arr[], int n)

    {

        if (n == 1)

            return true;

        // Sort array

        Arrays.sort(arr);

        // After sorting, difference between

        // consecutive elements must be same.

        int d = arr[1] - arr[0];

        for (int i = 2; i < n; i++)

            if (arr[i] - arr[i - 1] != d)

                return false;

        return true;

    }

    // driver code

    public static void main(String[] args)

    {

        int arr[] = { 20, 15, 5, 0, 10 };

        int n = arr.length;

        if (checkIsAP(arr, n))

            System.out.println("Yes");

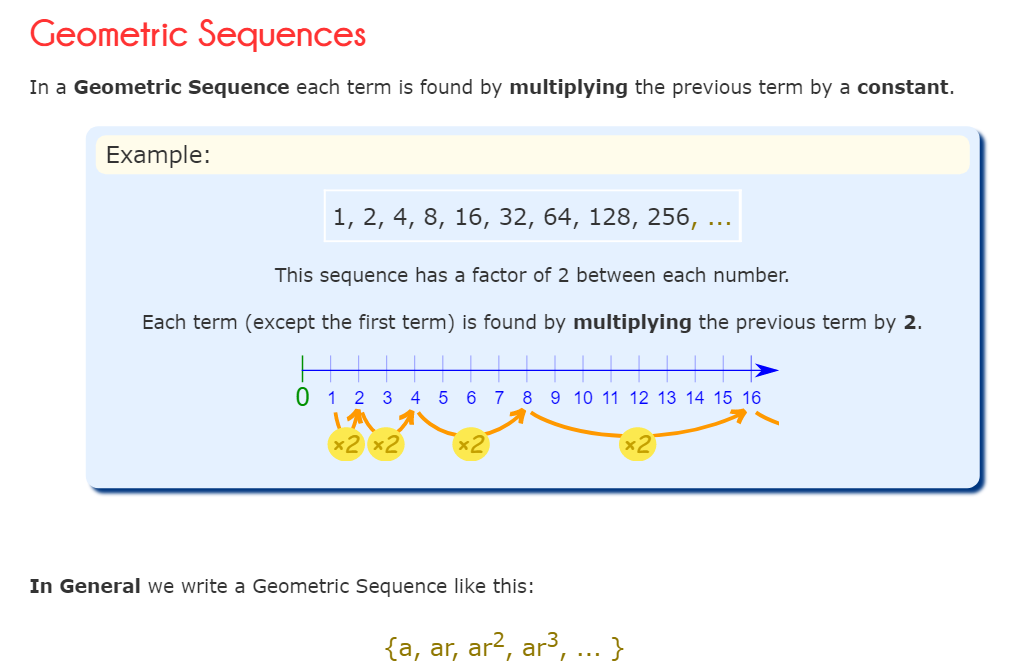
        else

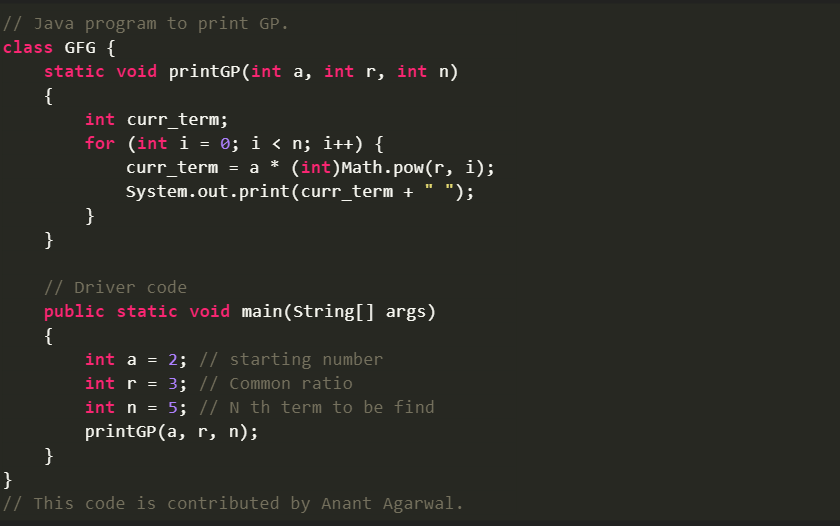
            System.out.println("No");

    }

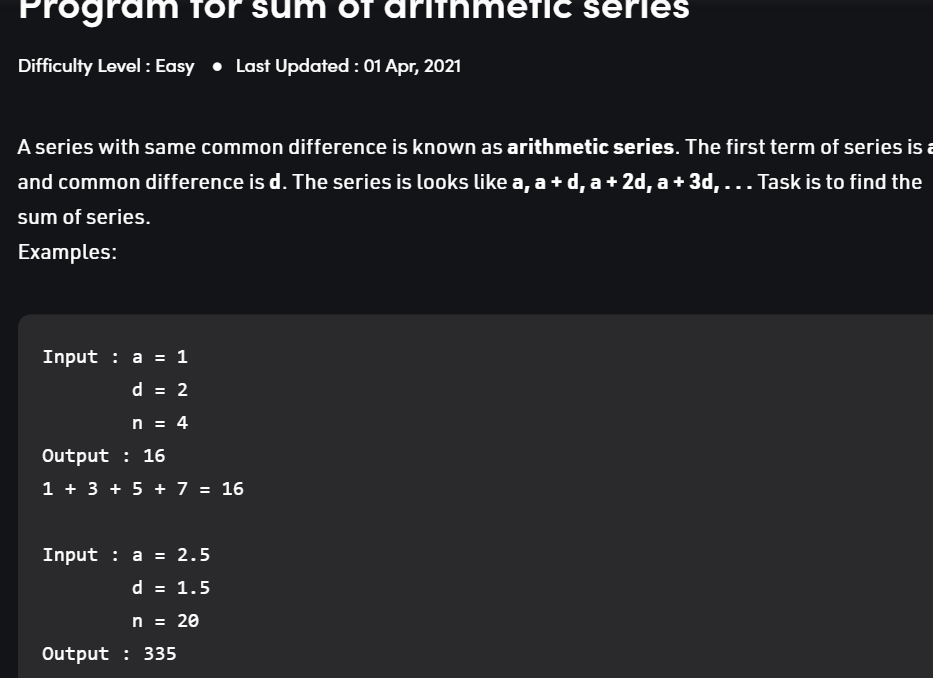
}

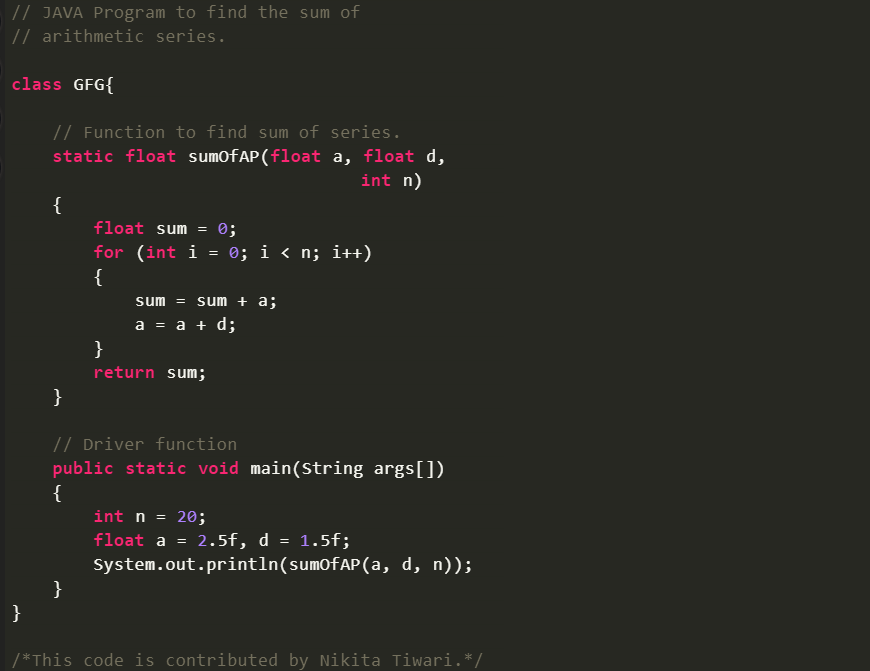
**29)Geometric sequence (1^2,2^2,3^2)**





**30)How to do a sum of numbers (foe loop/arithmetic sequence)**





**31)Binary Numbers**

1011101111 - use calculator

**32)Hexadecimal Numbers**

The hexadecimal numeral system, often shortened to "hex", is a **numeral system made up of 16 symbols (base 16) 0-15**

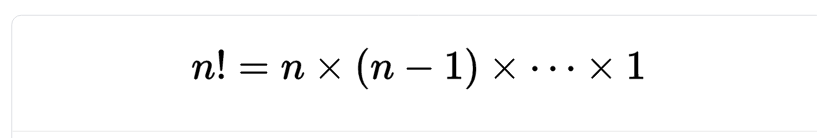
Use calculator

**33)Octal Numbers (8)**

A number system **with its base as eight** and uses digits from 0 to 7 is called Octal Number System. The word octal is used to represent the numbers that have eight as the base. The octal numbers have many applications and importance such as it is used in computers and digital numbering systems.

Use Calculator

**34)Factorial**



**public** BigInteger **factorialHavingLargeResult**(**int** n) { BigInteger result = BigInteger.ONE; **for** (**int** i = 2; i <= n; i++) result = result.multiply(BigInteger.valueOf(i)); **return** result; }

**35)Functions**

https://www.javatpoint.com/java-math

**36)Recursive Functions (Fibonacci sequence, Factorial of a number)**

https://www.javatpoint.com/recursion-in-java

**37)Maximum value/Upper limit, Lower limit of certain bytes of integer (32 vs 64 bits)**

# **Signed and Unsigned Integers**

The XDR standard defines signed integers as integer. A signed integer is a 32-bit datum that encodes an integer in the range [-2147483648 to 2147483647]. An unsigned integer is a 32-bit datum that encodes a nonnegative integer in the range [0 to 4294967295].

The signed integer is represented in twos complement notation. The most significant byte is 0 and the least significant is 3.

The unsigned integer is represented by an unsigned binary number whose most significant byte is 0; the least significant is 3. See the Signed Integer and Unsigned Integer figure ([Figure 1](https://www.ibm.com/docs/en/aix/7.1?topic=types-signed-unsigned-integers#a318c91d58__signintegunsiginteg)).

**public** **class** **Test** { **public** **static** **void** **main**(String[] args) { System.out.println(Integer.MIN\_VALUE); System.out.println(Integer.MAX\_VALUE); } }

**How much can 64-bit handle?**

**In principle, a 64-bit microprocessor can address 16 EiB (16 × 10246 = 264 = 18,446,744,073,709,551,616 bytes, or about 18.4 exabytes) of memory. However, not all instruction sets, and not all processors implementing those instruction sets, support a full 64-bit virtual or physical address space.**

**38)Floating point exceptions (Epsilon, division by 0, Infinity)**

# **Floating-point exceptions**

This topic provides information about floating-point exceptions and how your programs can detect and handle them.

The Institute of Electrical and Electronics Engineers (IEEE) defines a standard for floating-point exceptions called the IEEE Standard for Binary Floating-Point Arithmetic (IEEE 754). This standard defines five types of floating-point exception that must be signaled when detected:

* Invalid operation
* Division by zero
* Overflow
* Underflow
* Inexact calculation

When one of these exceptions occurs in a user process, it is signaled either by setting a flag or taking a trap. By default, the system sets a status flag in the Floating-Point Status and Control registers (FPSCR), indicating the exception has occurred. Once the status flags are set by an exception, they are cleared only when the process clears them explicitly or when the process ends. The operating system provides subroutines to query, set, or clear these flags.

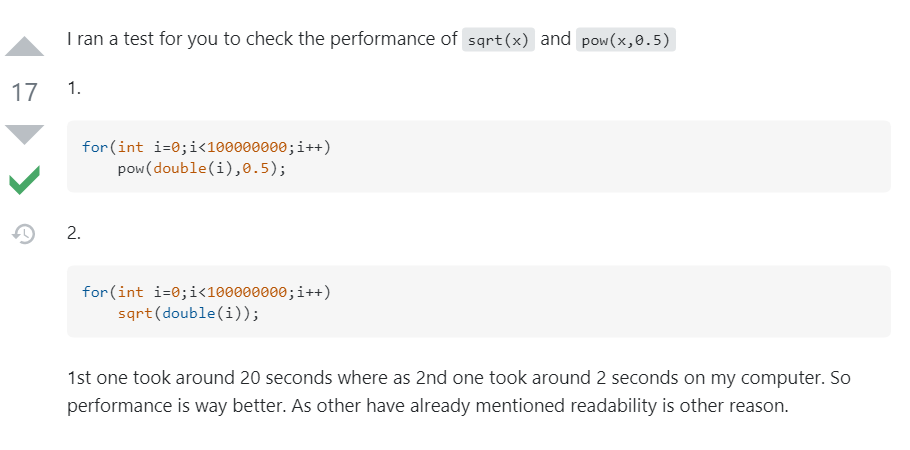
The system can also cause the floating-point exception signal (**SIGFPE**) to be raised if a floating-point exception occurs. Because this is not the default behavior, the operating system provides subroutines to change the state of the process so the signal is enabled. When a floating-point exception raises the **SIGFPE** signal, the process terminates and produces a core file if no signal-handler subroutine is present in the process. Otherwise, the process calls the signal-handler subroutine.

**Floating-point exception subroutines**

Floating-point exception subroutines can be used to:

* Change the execution state of the process
* Enable the signaling of exceptions
* Disable exceptions or clear flags
* Determine which exceptions caused the signal
* Test the exception sticky flags
* **Floating-point trap handler operation**
* To generate a trap, a program must change the execution state of the process using the **fp\_trap** subroutine and enable the exception to be trapped using the **fp\_enable** or **fp\_enable\_all** subroutine.
* Changing the execution state of the program may slow performance because floating-point trapping causes the process to execute in serial mode.
* When a floating-point trap occurs, the **SIGFPE** signal is raised. By default, the **SIGFPE** signal causes the process to terminate and produce a core file. To change this behavior, the program must establish a signal handler for this signal. See the **[sigaction](https://www.ibm.com/docs/en/ssw_aix_71/s_bostechref/sigaction.html)**, **[sigvec](https://www.ibm.com/docs/en/ssw_aix_71/s_bostechref/sigaction.html)**, or [**signal**](https://www.ibm.com/docs/en/ssw_aix_71/s_bostechref/sigaction.html) subroutines for more information on signal handlers.

**39)Square Root(sqrt/pow(x,1/2)**



**40)String (A set of 8bit numbers)**

**UTF-8** is a [variable-width](https://en.wikipedia.org/wiki/Variable-width_encoding) [character encoding](https://en.wikipedia.org/wiki/Character_encoding) used for electronic communication. Defined by the Unicode Standard, the name is derived from *Unicode* (or *Universal Coded Character Set*) *Transformation Format – 8-bit*.[[1]](https://en.wikipedia.org/wiki/UTF-8#cite_note-1)

**41) Set**

Set operations is **a concept similar to fundamental operations on numbers**. Sets in math deal with a finite collection of objects, be it numbers, alphabets, or any real-world objects. Sometimes a necessity arises wherein we need to establish the relationship between two or more sets.

