**BIG UNSIGNED INT CLASS – Documentation**

This is a custom variable type for unsigned int up to 10049 digits long. It doesn’t support negative numbers or float types.

Cap. ! – Initialization/Declaration

BIG UNSIGNED INT variables can be initialized from int, long long or char array types.

Example: BIGUNSIGNEDINT a(5), b(123456789123456789), c(“1234567890123456789012345678”);

For int and long longs, ‘=’ can also be used (BIGUNSIGNEDINT a=5), but for char arrays the ‘set()’ function needs to be used (b[]=”12345678901234567890”; a.set(b)). The function ‘set\_zero()’ sets a number to 0 (a.set\_zero()).

Cap II – Mathematical and Logic Operations

BIGUNSIGNEDINT variables can be used to perform mathematical or logical operations with BIGUNSIGNEDINT, int or long long types.

Example: BIGUNSIGNEDINT a; a=a+5; a=a\*a; if(a>123456); if(a!=34567); bigpow(a, 5); bigpow(5, a); factorial(a) -> Max 3400; prime(a); sum\_digit(a), bigsqrt(a); big\_cbrt(a);

Cap III – Printing and Input

You can easily print or read BIGUNSIGNEDINT variables with the << and >> operator, just like any other variable. Additionally, there is the format\_print(a, type) function that adds a thousand’s separator. Type indicates the type of separator.

//type 1 -> '

//type 2 -> .

//type 3 -> ,

//type 4 -> `

//type 5 -> space ' '

Example: format\_print(a, 2) -> 1.234.456.789

Cap IV – Other Functions

If your BIGUNSIGNEDINT variable is within the limits, you can convert it to int or long long types. Moreover there is a Random Number Generator, bigranrange(seed, min, max).

seed -> random number chosen by whoever uses this function

min -> a minimum value

max -> a maximum value