Contents

Chapter 1

Functions

1.1 gcd – gcd algorithm

1.1.1 gcd – the greatest common divisor

gcd(a: *integer*, b: *integer*) → *integer*

Return the greatest common divisor of two integers a and b.

a, b must be int, long or **Integer**. Even if one of the arguments is negative, the result is non-negative.

1.1.2 binarygcd – binary gcd algorithm

binarygcd(a: *integer*, b: *integer*) → *integer*

Return the greatest common divisor of two integers a and b by binary gcd algorithm.

†This function is an alias of **binarygcd**

a, b must be int, long, or **Integer**.

1.1.3 extgcd – extended gcd algorithm

extgcd(a: *integer*, b: *integer*) → (*integer*, *integer*, *integer*)

Return the greatest common divisor d of two integers a and b and u, v such that $d = \mathtt{a}u + \mathtt{b}v$.

`a`, `b` must be int, long, or **Integer**.
The returned value is a tuple (`u`, `v`, `d`).

1.1.4 lcm – the least common multiple

lcm(`a`: *integer*, `b`: *integer*) → *integer*

Return the least common multiple of two integers `a` and `b`.

†If both `a` and `b` are zero, then it raises an exception.
`a`, `b` must be int, long, or **Integer**.

1.1.5 gcd_of_list – gcd of many integers

gcd_of_list(`integers`: *list*) → *list*

Return gcd of multiple integers.

For given `integers` $[x_1, \ldots, x_n]$, return a list $[d,\ [c_1, \ldots, c_n]]$ such that $d = c_1 x_1 + \cdots + c_n x_n$, where d is the greatest common divisor of $x_1, \ldots, x_n$.

`integers` is a list which elements are int or long
This function returns $[d,\ [c_1, \ldots, c_n]]$, where $d,\ c_i$ are an integer.

1.1.6 coprime – coprime check

coprime(`a`: *integer*, `b`: *integer*) → *bool*

Return True if `a` and `b` are coprime, False otherwise.

`a`, `b` are int, long, or **Integer**.

1.1.7 pairwise_coprime – coprime check of many integers

pairwise_coprime(`integers`: *list*) → *bool*

Return True if all integers in `integers` are pairwise coprime, False otherwise.

`integers` is a list which elements are int, long, or **Integer**.

Examples

```
>>> gcd.gcd(12, 18)
6
>>> gcd.gcd(12, -18)
6
>>> gcd.gcd(-12, -18)
6
>>> gcd.extgcd(12, -18)
(-1, -1, 6)
>>> gcd.extgcd(-12, -18)
(1, -1, 6)
>>> gcd.extgcd(0, -18)
(0, -1, 18)
>>> gcd.lcm(12, 18)
36
>>> gcd.lcm(12, -18)
-36
>>> gcd.gcd_of_list([60, 90, 210])
[30, [-1, 1, 0]]
```

Bibliography