Prac 5 Design.

map

MapGeneric:

Member Variable:

vector<int> &map(std::vector<int>

is the vector for map input

vector<int> mapv {}

is the vector variable for store the output after mapping

Method:

virtual f(int) =0 is Pure virtual function, for children class

recusionmap (vector<int>, int) is for recusion map.

MapSquare:

Mememberr variable:

int i is integer variable ~~from~~ from input vector

Method:

MapSquare::f(int i) is function for $i \to i^2$.

MapTriple:

Mememberr variable:

int i is integer variable from input vector.

Method:

MapTriple::f(int i) is function for $i \to 3*i$

MapAbsolutable:

Memember variable:

int i is integer variable from input vector

Method:

MapAbsolute: f(int i) is a functn for input → |input|
abs ↑

filter

FilterGeneric:

memember variable:

vector<int>& filter(vector<int>).

is the vector for filter input

vector<int> filteru

is the vector for store the output after filtering

Method

virtual g(int) =0 is a pure virtual functn for children class

recursionfilter (vector<int>, int> is for recursion filter

FilterOdd:

Memember variable

int i is integer variable from input vector

Method:

Filter Square: f(int i) is a functn for return i. → i%2 != 0

FilterNonPositive:

Memember Variable.

int i is integer variable fm input vector.

Method:

FilterNopositive: g(int i) is return the integer >0.

FilterForTwoDigitPosit

Memember:

int i is a integer variable fn input vector.

Method:

FilterForADigitPositi: g(int i) is return the integer >10 & >0.

Reduce

ReduceGeneric

Memember variable:

int reduce(vector<int>) is the integer for input vector.

int reduceV is for store the output

Method:

recursivereduce() for recusion Reduce

virtual binarOperator(int, int) =0 for children class

ReduceGCD

Memember variable

i&j are integer variable for 2 number seek GCD.

Method.

ReduceGCD::binaryOperator for output return GCD

ReduceMininum

Memember variable

i&j are integer variable for 2 number seek mini one

Method:

ReduceGCD::binaryOperator for output return mini one.

Testing:

input:

-5, -24, -123, -81, 205, 157, 84, 67, -83, -60,

-72, 192, -25, -20, -50, -181, -70, -15, -108, 423

output:

15 15

input: 0, 0, 0, 0

output:

NAL

input: 0 ... 0
n

output: NULL

input: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18. 19 20. 0. (No Comma).

output: NULL

input:

9, -1, 53, -16, 73, 128, 105, 104, -71, -179, 102, 12, 21, -15, -9, 179, -156, -186, 43, -189

output:

27.9