1. Compare the triplesst

$handle = fopen ("php://stdin", "r");

function solve($a0, $a1, $a2, $b0, $b1, $b2){

// Complete this function

$aScore = 0;

$bScore = 0;

$aScore += ($a0 > $b0? 1:0) + ($a1 > $b1? 1:0) + ($a2 > $b2? 1:0);

$bScore += ($b0 > $a0? 1:0) + ($b1 > $a1? 1:0) + ($b2 > $a2? 1:0);

$score = [];

array\_push($score,$aScore,$bScore);

return $score;

}

fscanf($handle, "%d %d %d", $a0, $a1, $a2);

fscanf($handle, "%d %d %d", $b0, $b1, $b2);

$result = solve($a0, $a1, $a2, $b0, $b1, $b2);

echo implode(" ", $result)."\n";

1. Very big sum

<?php

function aVeryBigSum($ar) {

$sum = 0;

foreach($ar as $value){

$sum+=$value;

}

return $sum;

}

1. Diagonal

function diagonalDifference($arr) {

$panjang = count($arr);

$diagonal\_Utama = 0;

$diagonal\_Kedua = 0;

$final\_Index = $panjang - 1;

for($f = 0; $f<$panjang; $f++){

$diagonal\_Utama+=$arr[$f][$f];

$diagonal\_Kedua+=$arr[$f][$final\_Index--];

}

return abs($diagonal\_Utama - $diagonal\_Kedua);

}

1. Max-min

// Complete the miniMaxSum function below.

function miniMaxSum($arr) {

//5 inputan (0-4)

$maxSum\_Arry = $arr;

$minSum\_Arry = $arr;

// asc sorting dan get value pertama

sort($minSum\_Arry);

$minjmlh = array\_splice($minSum\_Arry, 0 ,4);

// desc sorting dan get value pertama

rsort($maxSum\_Arry);

$maxjmlh = array\_splice($maxSum\_Arry, 0, 4);

echo array\_sum($minjmlh) . ' ' . array\_sum($maxjmlh);

}

1. Candle

// hitung jumlah candle

$initiate\_candle = array\_count\_values($arr);

// sort candle asc dan get yang paling bnyak candle pada line tertentu

krsort($initiate\_candle);

$canlde\_total = array\_values($initiate\_candle);

return $canlde\_total[0];

1. First solve
2. Plus minus

$positif = 0;

$negatif = 1;

$main = 2;

$panjang = count($val);

$angka = [0, 0, 0];

// determinasi angka plus minus

foreach($val as $int){

if($int === 0){

$angka[$main] = $angka[$main] + 1;

continue;

}

if($int > 0){

$angka[$positif] = $angka[$positif] + 1;

continue;

}

$angka[$negatif] = $angka[$negatif] + 1;

}

$plusMinusInt = array\_map(function ($angka) use ($panjang) {

return number\_format($angka/$panjang, 6);

}, $angka);

foreach($plusMinusInt as $final){

echo $final;

echo "\n";

}

}

1. Apel dan jeruk

$handle = fopen ("php://stdin","r");

fscanf($handle,"%d %d",$awal,$akhir);

fscanf($handle,"%d %d",$jarakA,$jarakB);

fscanf($handle,"%d %d",$x,$y);

$apel\_tot = fgets($handle);

$apel = explode(" ",$apel\_tot);

array\_walk($apel,'intval');

$jeruk\_tot = fgets($handle);

$jeruk = explode(" ",$jeruk\_tot);

array\_walk($jeruk,'intval');

$final\_A = 0; $final\_B = 0;

for ($i = 0; $i < count($apel); $i++) {

$rest\_A = ($jarakA+$apel[$i]);

if ($rest\_A >= $awal && $rest\_A <= $akhir) {

$final\_A++;

}

}

echo $final\_A."\n";

for ($j = 0; $j < count($jeruk); $j++) {

$rest\_B = ($jarakB+$jeruk[$j]);

if ($rest\_B >= $awal && $rest\_B <= $akhir) {

$final\_B++;

}

}

echo $final\_B."\n";

1. Kanguru

if ($v1 <= $v2) {

echo "NO";

} else {

$kanguru = ($x2 - $x1) % ($v2 - $v1) == 0;

echo ($kanguru ? "YES" : "NO");

}

1. Breaking records

function breakingRecord($f){

// Complete this function

$rekorTertinggi = $f[0];

$rekorTerendah = $f[0];

$terbaik = 0;

$terburuk = 0;

$jmlh = count($f);

for($i=1;$i<$jmlh;$i++){

if($f[$i] > $rekorTertinggi){

$rekorTertinggi = $f[$i];

$terbaik++;

}

if($f[$i] < $rekorTerendah){

$rekorTerendah = $f[$i];

$terburuk++;

}

}

return array($terbaik,$terburuk);

}

fscanf($handle,"%d",$n);

$f\_temp = fgets($handle);

$f = explode(" ",$f\_temp);

$f = array\_map('intval', $f);

$hasil = breakingRecord($f);

echo implode(" ", $hasil)."\n";

1. Encryption (medium)

$dataA = floor(sqrt(strlen($f)));

$dataB = ceil(sqrt(strlen($f)));

$val = str\_split($f, $dataB);

$hasil = [];

for ($i = 0; $i < $dataB; $i ++) {

$baru = '';

for ($j = 0; $j <= $dataA; $j ++) {

if (isset($val[$j][$i])) {

$baru .= $val[$j][$i];

}

}

$hasil[] = $baru;

}

echo implode(" ", $hasil);