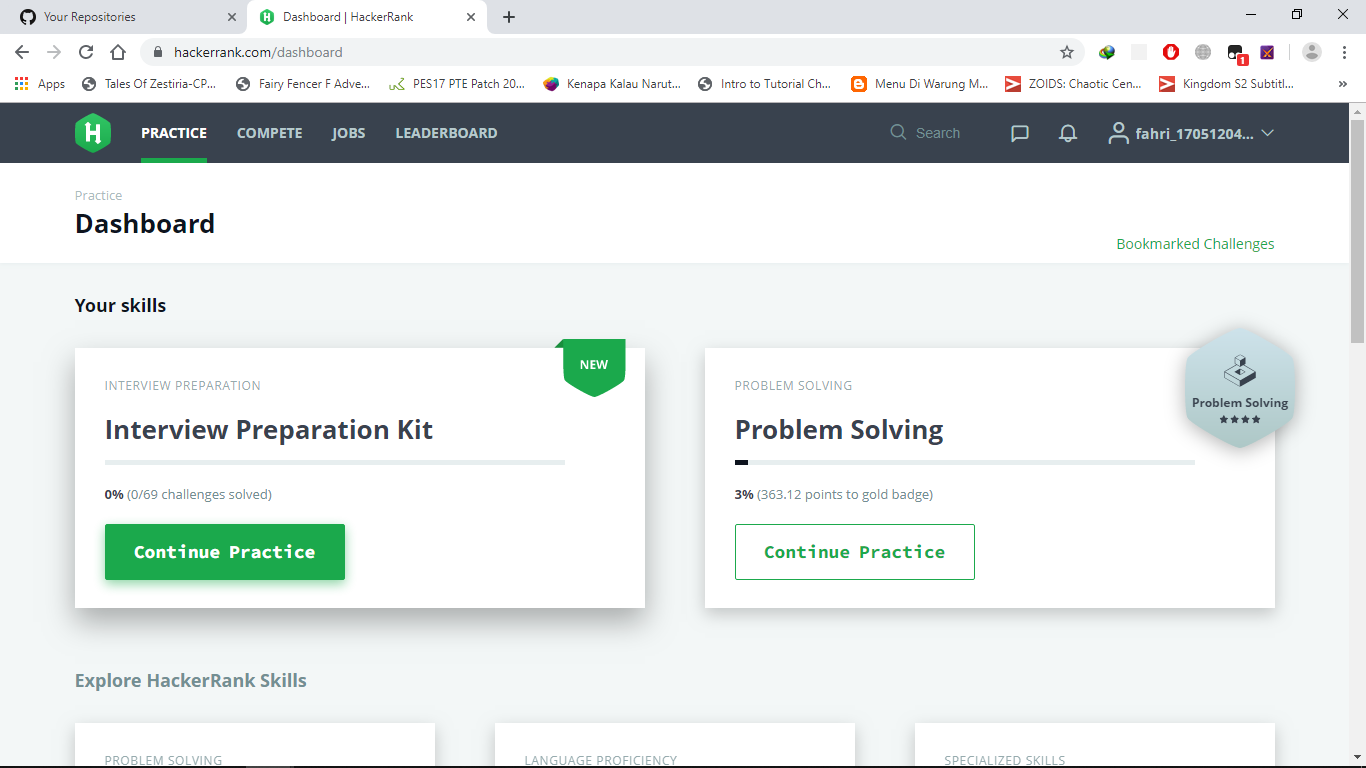
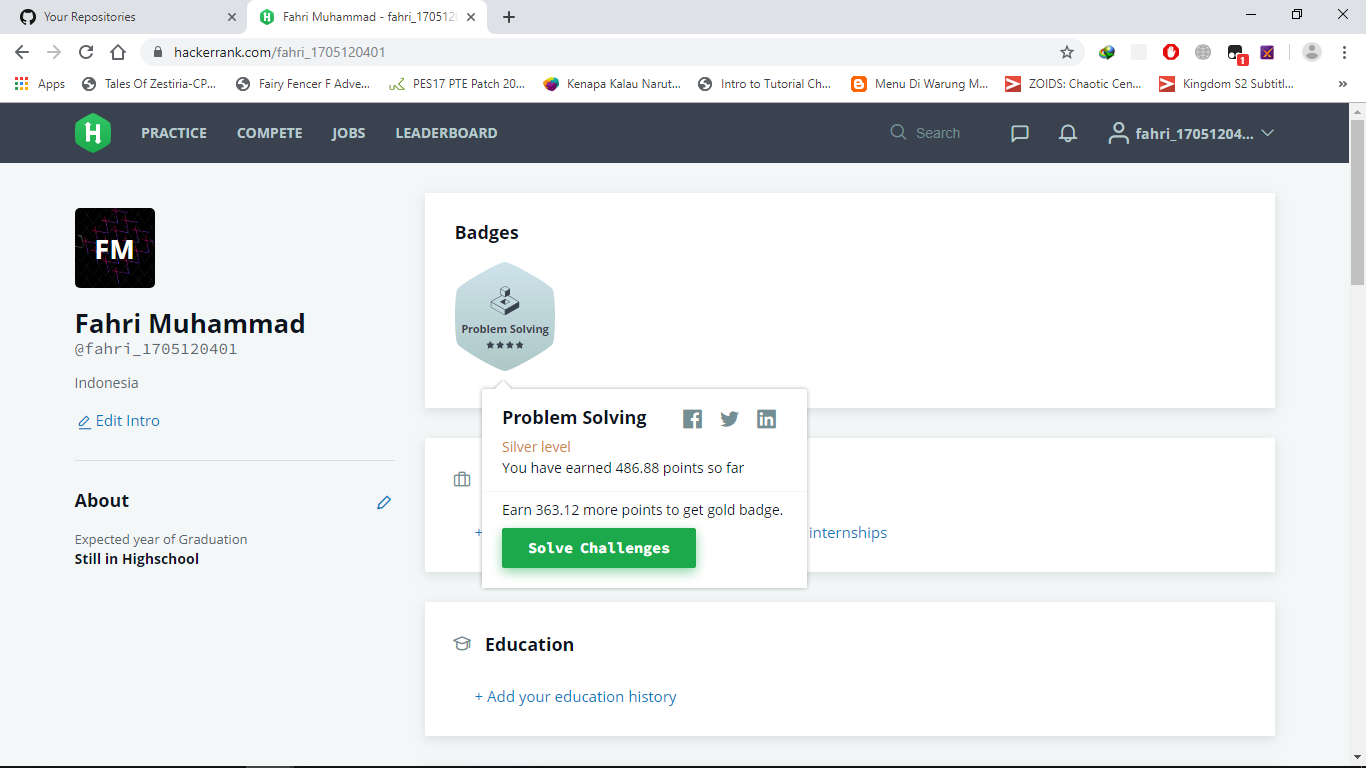
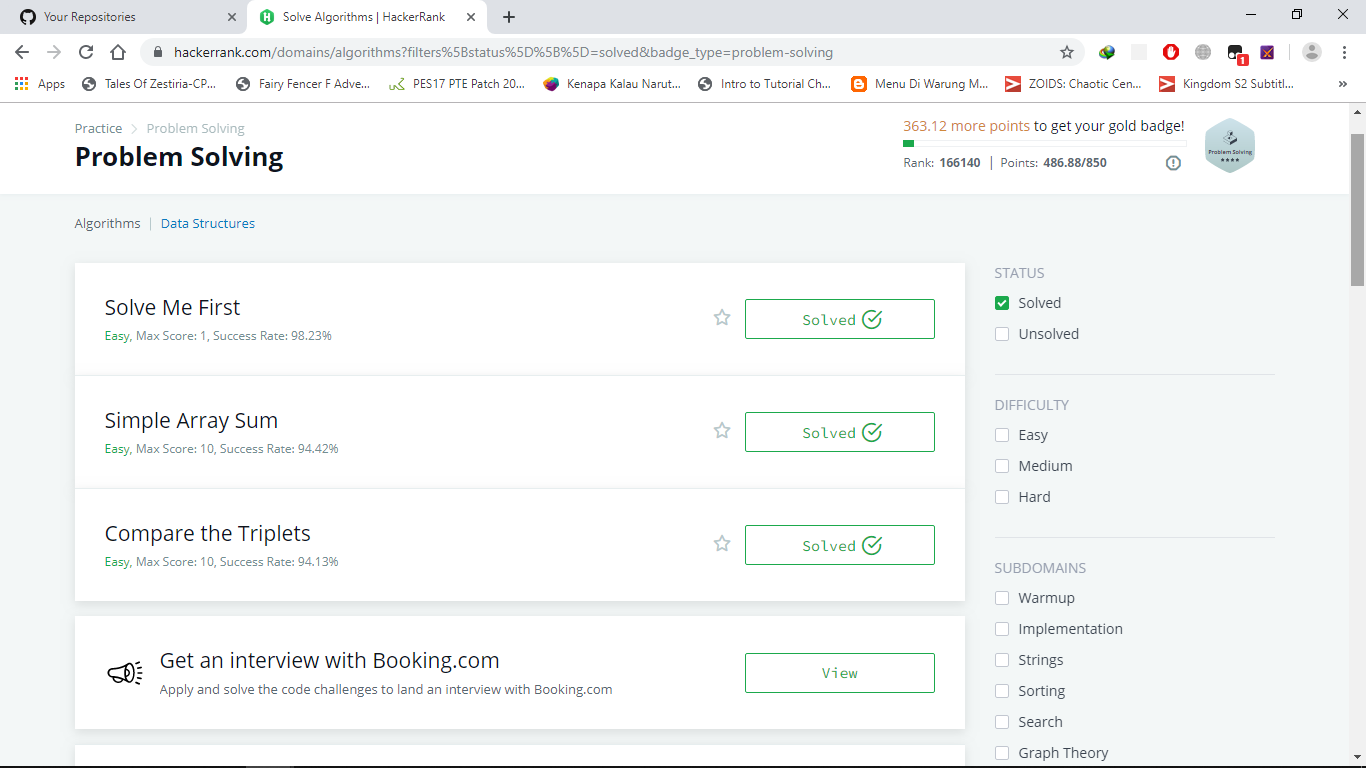
Nama : Fahri Muhammad

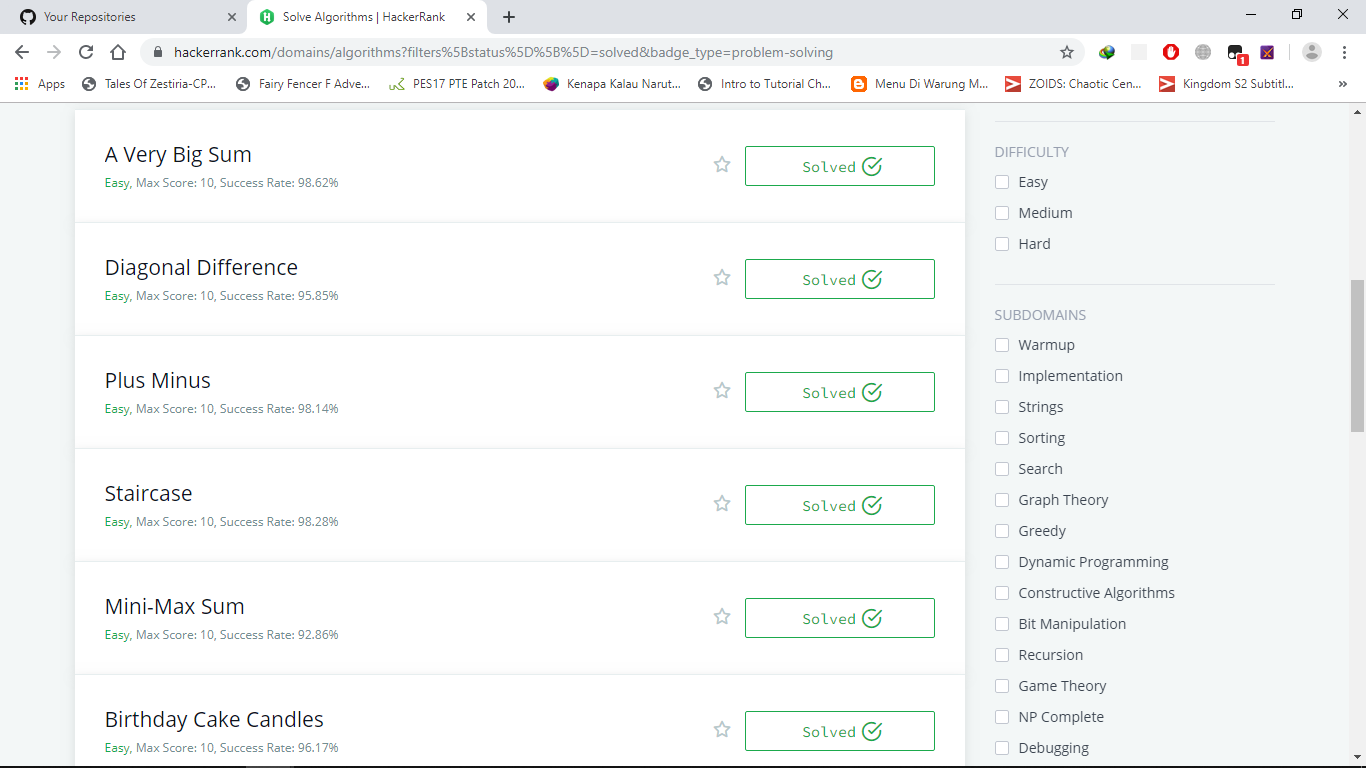
Nim : 17051204029

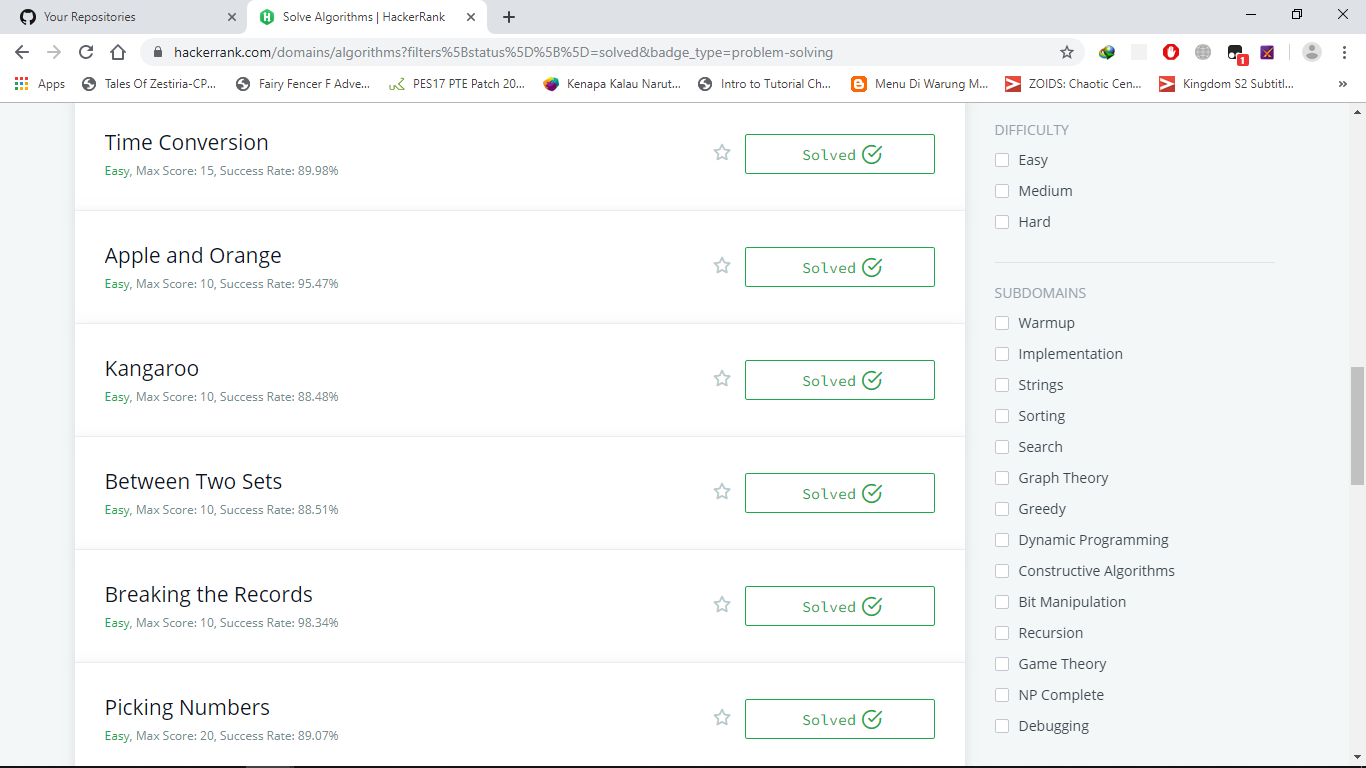
Kelas : TI 2017 A

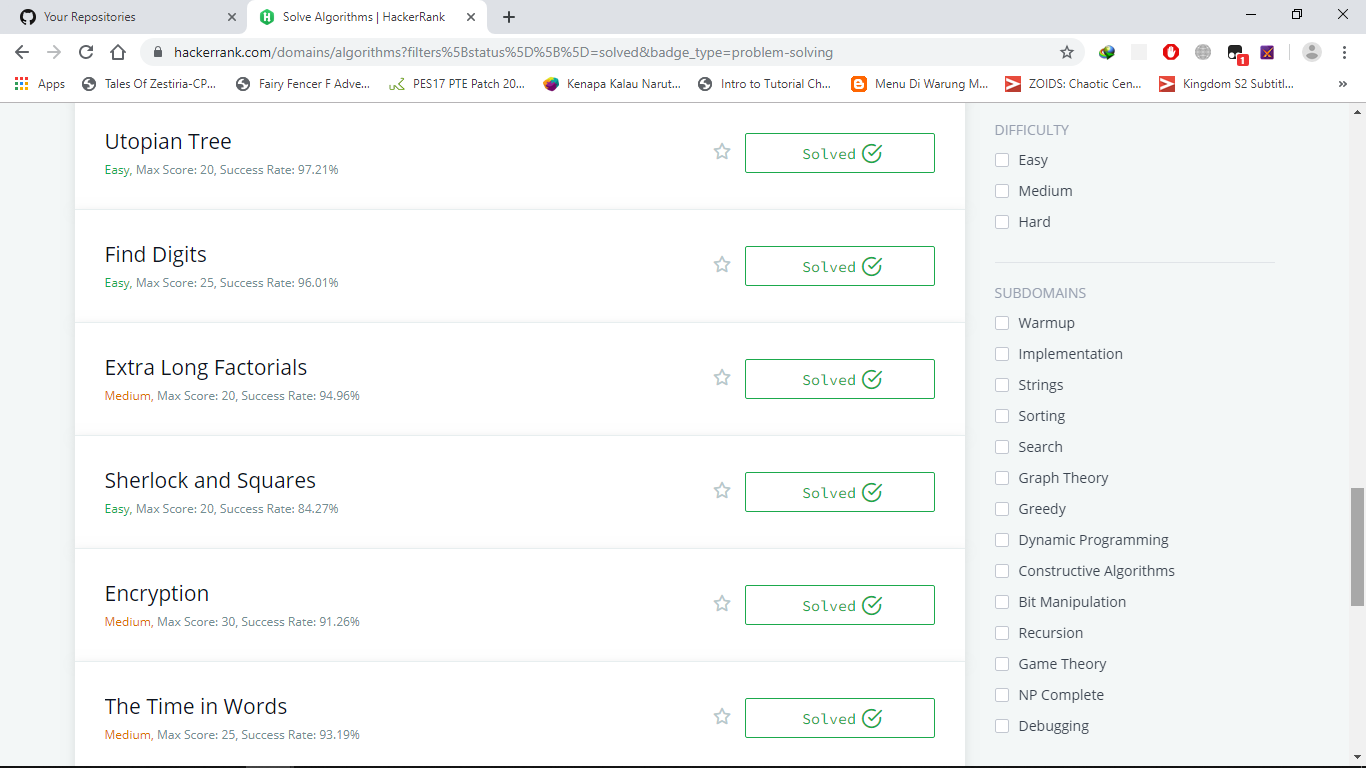


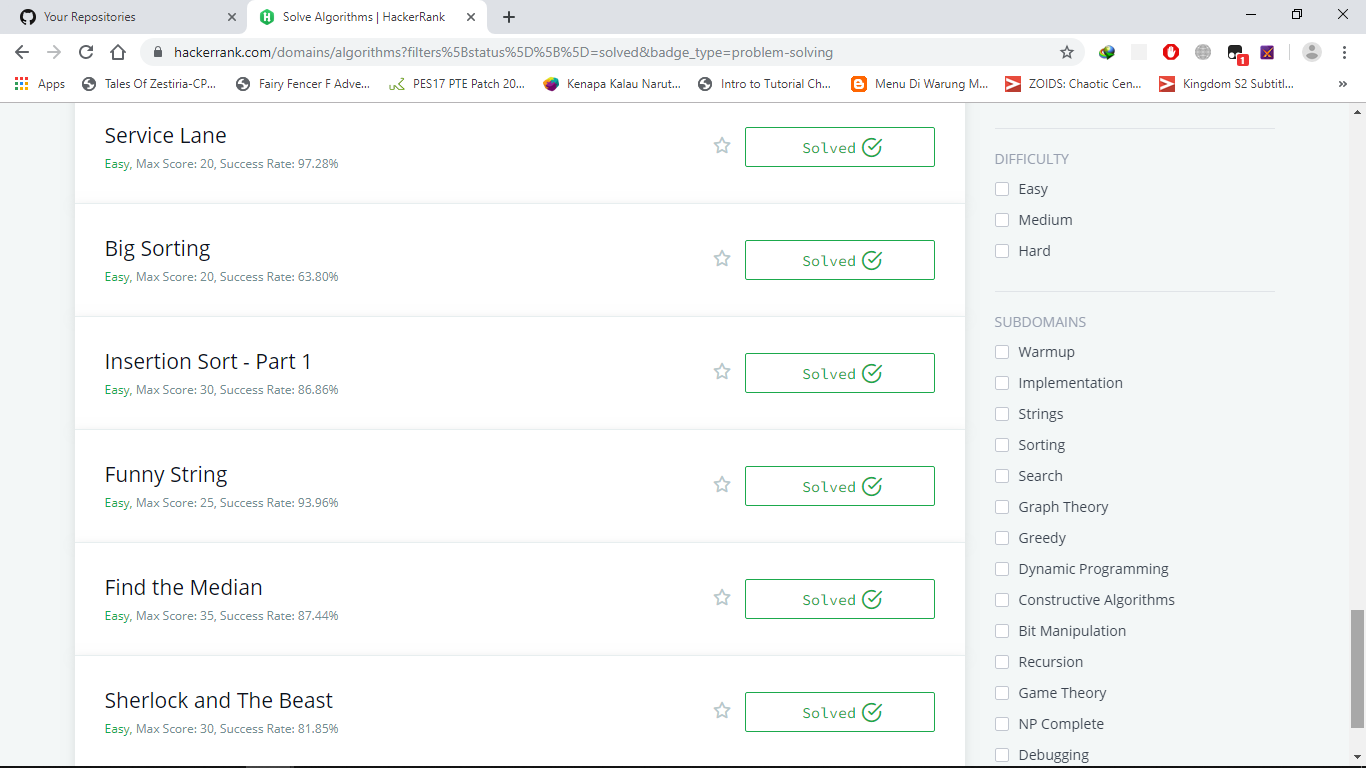












# Original Code (PHP)

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 Difference

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. Mini-max sum

// 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. Birthday Cake 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. Solve me First

<?php

function solveMeFirst($a,$b){

// Hint: Type return $a + $b; below

}

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

$\_a = fgets($handle);

$\_b = fgets($handle);

$sum = solveMeFirst((int)$\_a,(int)$\_b);

print ($sum);

fclose($handle);

?>

1. 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. Apple and Orange

$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. Kangaroo

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);

1. Simple Array Sum

<?php

/\*

\* Complete the simpleArraySum function below.

\*/

function simpleArraySum($val) {

$sum = 0;

foreach($val as $value){

$sum+=$value;

}

return $sum;

}

$fptr = fopen(getenv("OUTPUT\_PATH"), "w");

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

fscanf($stdin, "%d\n", $ar\_count);

fscanf($stdin, "%[^\n]", $ar\_temp);

$val = array\_map('intval', preg\_split('/ /', $ar\_temp, -1, PREG\_SPLIT\_NO\_EMPTY));

$result = simpleArraySum($val);

fwrite($fptr, $result . "\n");

fclose($stdin);

fclose($fptr);

1. Very Big Sum

<?php

// Complete the aVeryBigSum function below.

function aVeryBigSum($ar) {

$sum = 0;

foreach($ar as $value){

$sum+=$value;

}

return $sum;

}

$fptr = fopen(getenv("OUTPUT\_PATH"), "w");

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

fscanf($stdin, "%d\n", $ar\_count);

fscanf($stdin, "%[^\n]", $ar\_temp);

$ar = array\_map('intval', preg\_split('/ /', $ar\_temp, -1, PREG\_SPLIT\_NO\_EMPTY));

$result = aVeryBigSum($ar);

fwrite($fptr, $result . "\n");

fclose($stdin);

fclose($fptr);

1. Staircase

<?php

// Complete the staircase function below.

function staircase($f) {

$langkah\_Total = $f;

$staircase = [];

// perulangan dari kanan

for($f; $f>0; $f--){

$stairs = [];

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

if($i == $f){

// hitung total

$total\_stair = ($langkah\_Total - $f) + 1;

for($total\_stair; $total\_stair>0; $total\_stair--){

array\_push($stairs, "#");

}

array\_push($staircase, $stairs);

break;

}

array\_push($stairs, " ");

}

}

// print the staircases

foreach($staircase as $stairs){

foreach($stairs as $stair){

echo $stair;

}

echo "\n";

}

}

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

fscanf($stdin, "%d\n", $f);

staircase($f);

fclose($stdin);

1. Time Convertions

<?php

/\*

\* Complete the timeConversion function below.

\*/

function timeConversion($f) {

$time = strtotime($f);

return date('H:i:s', $time);

}

$fptr = fopen(getenv("OUTPUT\_PATH"), "w");

$\_\_fp = fopen("php://stdin", "r");

fscanf($\_\_fp, "%[^\n]", $s);

$result = timeConversion($s);

fwrite($fptr, $result . "\n");

fclose($\_\_fp);

fclose($fptr);

1. Beetwen Two Sets

<?php

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

fscanf($stdin,"%d %d",$f,$m);

$x\_temp = fgets($stdin);

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

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

$y\_temp = fgets($stdin);

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

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

$y\_Min=(int)min($y); $x\_Max=(int)max($x); $jumlah=0;

for ($i = $x\_Max; $i <= $y\_Min; $i++) {

$z = true;

for ($j = 0; $j < $f; $j++){ if ($i%$x[$j]!=0) { $z = false; break; } }

for ($j = 0; $j < $m; $j++){ if ($y[$j]%$i!=0) { $z = false; break; } }

if ($z == true) $jumlah++;

}

echo $jumlah."\n";

1. Picking Numbers

<?php

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

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

$x\_temp = fgets($stdin);

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

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

$x = array\_count\_values($x);

$num = -1;

for($i = 1; $i <99; $i++){

$num = max($num, $x[$i] + $x[$i+1]);

}

echo $num;

1. Utopian Tree

<?php

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

fscanf($stdin,"%d",$f);

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

fscanf($stdin,"%d",$m);

$h = 1;

for ($i = 0; $i < $m; $i++) $i % 2 == 0 ? $h \*= 2 : $h += 1;

echo $h."\n";

}

1. Find Digits

<?php

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

fscanf($stdin, "%d", $f);

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

fscanf($stdin, "%d", $x);

$y = $x;

$hasil = 0;

while ($y > 0) {

$digit = $y % 10;

$y = (int) $y / 10;

if ($digit != 0 && $x % $digit == 0) {

$hasil ++;

}

}

echo $hasil . "\n";

}

1. Extra Long Factorial

<?php

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

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

// one

$result = 1;

while ($n >= 1) {

$result = bcmul($result, $n);

$n --;

}

echo $result . PHP\_EOL;

// two

$arr = [];

$arr[0] = 1;

$k = 0;

$carry = 0;

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

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

$arr[$j] = $arr[$j] \* $i + $carry;

$carry = $arr[$j] / 10;

$arr[$j] %= 10;

}

while ($carry) {

$k ++;

$arr[$k] = $carry % 10;

$carry /= 10;

}

}

1. Sherlock and Square

<?php

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

fscanf($stdin, "%d", $f);

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

fscanf($stdin, "%d %d", $a, $b);

echo floor(sqrt($b)) - ceil(sqrt($a)) + 1;

echo"\n";

}

1. The time in Words

<?php

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

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

fscanf($handle,"%s",$m);

$wkt = [

1 => 'one',

2 => 'two',

3 => 'three',

4 => 'four',

5 => 'five',

6 => 'six',

7 => 'seven',

8 => 'eight',

9 => 'nine',

10 => 'ten',

11 => 'eleven',

12 => 'twelve',

13 => 'thirteen',

14 => 'fourteen',

15 => 'fifteen',

16 => 'sixteen',

17 => 'seventeen',

18 => 'eighteen',

19 => 'nineteen',

20 => 'twenty'

];

function \_next($first) {

return $first == 12 ? 1 : $first + 1;

}

$menit = function($first) use ($wkt) {

if ($first > 20) {

return $wkt[20] . ' ' . $wkt[(int) ($first % 10)];

}

return $wkt[$first];

};

if ($m == 0) {

echo $wkt[$h] . ' o\' clock' . PHP\_EOL;

} else if ($m == 1) {

echo $wkt[(int)$m] . ' minute past ' . $wkt[$h] . PHP\_EOL;

} else if ($m == 15) {

echo 'quarter past ' . $wkt[$h] . PHP\_EOL;

} else if ($m < 30) {

echo $menit((int)$m) . ' minutes past ' . $wkt[$h] . PHP\_EOL;

} else if ($m == 30) {

echo 'half past ' . $wkt[$h] . PHP\_EOL;

} else if ($m == 31) {

echo $menit(60 - (int)$m) . ' minute to ' . $wkt[\_next($h)] . PHP\_EOL;

} else if ($m == 45) {

echo 'quarter to ' . $wkt[\_next($h)] . PHP\_EOL;

} else {

echo $menit(60 - (int)$m) . ' minutes to ' . $wkt[\_next($h)] . PHP\_EOL;

}

1. Service Lane

<?php

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

fscanf($stdin,"%d %d",$f,$m);

$\_lebar = fgets($stdin);

$lebar = explode(" ",$\_lebar);

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

for($x = 0; $x < $m; $x++){

fscanf($stdin,"%d %d",$i,$j);

$fnal = array\_slice($lebar, $i, $j - $i + 1);

printf("%d\n", min($fnal));

}

1. Big Sorting

<?php

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

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

$unsorted = array();

for($unsorted\_i = 0; $unsorted\_i < $n; $unsorted\_i++){

fscanf($handle,"%s",$unsorted[]);

}

function cmp($a, $b)

{

if(strlen($a) == strlen($b)){

$i = 0;

while($i < strlen($a)){

if($a[$i] != $b[$i]){

return $a[$i] > $b[$i];

}

$i++;

}

}else{

return (strlen($a) > strlen($b));

}

}

usort($unsorted, "cmp");

echo(implode("\n",$unsorted));

1. Insertion Sort – Part 1

<?php

function printar($val) {

foreach ($val as $x) echo $x." ";

echo "\n";

}

function insertionSort( $val) {

$last = $val[count($val) - 1];

$y = count($val)-2;

while ($y + 1 > 0 && $last <= $val[$y]) {

$val[$y+1] = $val[$y--];

printar($val);

}

$val[$y+1] = $last;

printar($val);

}

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

fscanf($stdin, "%d", $f);

$val = array();

$s=fgets($stdin);

$val = explode(" ", $s);

for($y=0;$y < count($val);$val[$y++]+=0);

insertionSort($val);

1. Funny String

<?php

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

fscanf($stdin, "%d", $f);

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

fscanf($stdin, "%s", $m);

$res = 0;

for ($j = 0; $j < strlen($m)/2; $j++)

if (abs(ord($m[$j])-ord($m[$j+1])) - abs(ord($m[strlen($m)-$j-1])-ord($m[strlen($m)-$j-2]))) break;

if ($j < strlen($m)/2) echo "Not ";

echo "Funny\n";

}

1. Find the Median

<?php

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

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

$arr = explode(' ', trim(fgets($handle)));

sort($arr);

echo $arr[floor($n / 2)];

1. Sherlock and The Beast

<?php

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

fscanf($stdin,"%d",$f);

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

fscanf($stdin,"%d",$y);

$hsil = false;

for ($i = $y ; $i >= 0; $i -= 5) {

if (($y - $i) % 5 == 0 && ($i % 3 == 0)) {

echo str\_repeat('5', $i) . str\_repeat('3', $y - $i);

echo "\n";

$hsil = true;

break;

}

}

if (! $hsil) echo '-1'."\n";

}