<?php

$sure\_baslangici = microtime(true);

$sayi=rand(1,10);

$sayac=1;

$sayilar[]=$sayi;

while($sayac<10){

$sayi=rand(1,100);

$varmi=false; // her dizi elemanı üretilen sayı ile karşılaştırılacak

$indis=0;

while(@$sayilar[$indis]){

if ($sayilar[$indis]==$sayi) //eğer üretilen sayı dizide varsa

$varmi=true; // dizide var olarak işaretle

$indis++;

}

if ($varmi==false){ // eğer üretilen sayı dizide yoksa

$sayilar[]=$sayi; // diziye ekle

$sayac++;

}

}

$indis=0;

while(@$sayilar[$indis]){

echo $sayilar[$indis]," ";

$indis++;

}

function selection\_sort($sayilar)

{

for($i=0; $i<count($sayilar)-1; $i++) {

$min = $i;

for($j=$i+1; $j<count($sayilar); $j++) {

if ($sayilar[$j]<$sayilar[$min]) {

$min = $j;

}

}

$sayilar = swap\_positions($sayilar, $i, $min);

}

return $sayilar;

}

function swap\_positions($data1, $left, $right) {

$backup\_old\_data\_right\_value = $data1[$right];

$data1[$right] = $data1[$left];

$data1[$left] = $backup\_old\_data\_right\_value;

return $data1;

}

echo "\nOriginal Array :\n";

echo implode(', ',$sayilar );

echo "\nSorted Array :\n";

echo implode(', ',selection\_sort($sayilar)). PHP\_EOL;

$sure\_bitimi = microtime(true);

$sure = $sure\_bitimi - $sure\_baslangici;

echo "<br>Bekleme süresi: $sure saniye.\n";

//PHP kodlarına ayrılan belleğin miktarını bayt cinsinden döndürür.

echo 'Hafıza kullanımı: ',round(memory\_get\_peak\_usage()/1048576, 2), 'MB';

?>

<?php

$sure\_baslangici = microtime(true);

$sayi=rand(1,10);

$sayac=1;

$sayilar[]=$sayi;

for($i=0; $i<10; $i++){

$sayi=rand(1,100);

$varmi=false; // her dizi elemanı üretilen sayı ile karşılaştırılacak

$indis=0;

while(@$sayilar[$indis]){

if ($sayilar[$indis]==$sayi) //eğer üretilen sayı dizide varsa

$varmi=true; // dizide var olarak işaretle

$indis++;

}

if ($varmi==false){ // eğer üretilen sayı dizide yoksa

$sayilar[]=$sayi; // diziye ekle

$sayac++;

}

}

$indis=0;

while(@$sayilar[$indis]){

echo $sayilar[$indis]," ";

$indis++;

}

function selection\_sort($sayilar)

{

for($i=0; $i<count($sayilar)-1; $i++) {

$min = $i;

for($j=$i+1; $j<count($sayilar); $j++) {

if ($sayilar[$j]<$sayilar[$min]) {

$min = $j;

}

}

$sayilar = swap\_positions($sayilar, $i, $min);

}

return $sayilar;

}

function swap\_positions($data1, $left, $right) {

$backup\_old\_data\_right\_value = $data1[$right];

$data1[$right] = $data1[$left];

$data1[$left] = $backup\_old\_data\_right\_value;

return $data1;

}

echo "\nOriginal Array :\n";

echo implode(', ',$sayilar );

echo "\nSorted Array :\n";

echo implode(', ',selection\_sort($sayilar)). PHP\_EOL;

$sure\_bitimi = microtime(true);

$sure = $sure\_bitimi - $sure\_baslangici;

echo "<br>Bekleme süresi: $sure saniye.\n";

//PHP kodlarına ayrılan belleğin miktarını bayt cinsinden döndürür.

echo 'Hafıza kullanımı: ',round(memory\_get\_peak\_usage()/1048576, 2), 'MB';

?>

<?php

for ($i=0; $i<12; $i++) {

$my\_array[$i] = $i+1;

}

$Gecici = 0;

for($Dongu = 0; $Dongu < count($my\_array); $Dongu++)

{

for($Sirala = 0; $Sirala < count($my\_array) - 1; $Sirala++)

{

if($my\_array[$Sirala] < $my\_array[$Sirala + 1])

{

$Gecici = $my\_array[$Sirala + 1];

$my\_array[$Sirala + 1] = $my\_array[$Sirala];

$my\_array[$Sirala] = $Gecici;

}

}

}

foreach($my\_array as $Sayilar) echo $Sayilar." ";

?>

<?php

for ($i=0; $i<12; $i++) {

$my\_array[$i] = $i+1;

}

$Gecici = 0;

for($Dongu = 0; $Dongu < count($my\_array); $Dongu++)

{

for($Sirala = 0; $Sirala < count($my\_array) - 1; $Sirala++)

{

if($my\_array[$Sirala] < $my\_array[$Sirala + 1])

{

$Gecici = $my\_array[$Sirala + 1];

$my\_array[$Sirala + 1] = $my\_array[$Sirala];

$my\_array[$Sirala] = $Gecici;

}

}

}

foreach($my\_array as $Sayilar) echo $Sayilar." ";

?>

<?php

$sure\_baslangici = microtime(true);

function merge\_sort($my\_array){

if(count($my\_array) == 1 ) return $my\_array;

$mid = count($my\_array) / 2;

$left = array\_slice($my\_array, 0, $mid);

$right = array\_slice($my\_array, $mid);

$left = merge\_sort($left);

$right = merge\_sort($right);

return merge($left, $right);

}

function merge($left, $right){

$res = array();

while (count($left) > 0 && count($right) > 0){

if($left[0] > $right[0]){

$res[] = $right[0];

$right = array\_slice($right , 1);

}else{

$res[] = $left[0];

$left = array\_slice($left, 1);

}

}

while (count($left) > 0){

$res[] = $left[0];

$left = array\_slice($left, 1);

}

while (count($right) > 0){

$res[] = $right[0];

$right = array\_slice($right, 1);

}

return $res;

}

$test\_array = array(100, 54, 7, 2, 5, 4, 1);

echo "Original Array : ";

echo implode(', ',$test\_array );

echo "\nSorted Array :";

echo implode(', ',merge\_sort($test\_array))."\n";

$sure\_bitimi = microtime(true);

$sure = $sure\_bitimi - $sure\_baslangici;

echo "<br>Bekleme süresi: $sure saniye.\n";

//PHP kodlarına ayrılan belleğin miktarını bayt cinsinden döndürür.

echo 'Hafıza kullanımı: ',round(memory\_get\_peak\_usage()/1048576, 2), 'MB';

?>

<?php

$sayilar=array(4,5,8,9,9,2,2,5);

$toplam=0;

foreach($sayilar as $sayi){

$toplam+=$sayi;

}

echo "Toplam = $toplam";

?>