1. **要求在一组数中，插入一个新数，并维护原来的排序方式不变**

<?php

//1：要求在一组数中，插入一个新数，并维护原来的排序方式不变

function insertArr($arr,$val){

$pos=0;

if (sizeof($arr)==0) return array($val); //传入数组没有值

if (sizeof($arr)==1){ //传入数组个数为1

return array($val,$arr[0]);

}

$bfind=false;

if ($arr[0]>$arr[sizeof($arr) - 1]){

$bfind=true; //从大到小排序

}else{

$bfind=false;//从小到大排序

}

for($i=0;$i<sizeof($arr);$i++){ //找到要插入值应该插入的位置。

if ($bfind){

if ($arr[$i]<$val){

$pos=$i;

break;

}

}else{

if ($arr[$i]>$val){

$pos=$i;

break;

}

}

}

$left\_arr=array();

$right\_arr=array();

for ($i=0;$i<sizeof($arr);$i++){ //根据要插入的值把数组分左右两部分。

if ($i<$pos){

$left\_arr[]=$arr[$i];

}else{

$right\_arr[]=$arr[$i];

}

}

return array\_merge($left\_arr,array($val),$right\_arr);

}

//测试

$arr=array();

print\_r(insertArr($arr,10));echo "<br>";

$arr=array(1);

print\_r(insertArr($arr,10));echo "<br>";

$arr=array(1,3,8,12,13,15);

print\_r(insertArr($arr,10));echo "<br>";

$arr=array(31,23,18,12,11,1);

print\_r(insertArr($arr,10));echo "<br>";

?>

1. **用二分法在一个数组中查找你所需元素**

<?php

//二分法

function binarySearch($a, $val){

$low = 0;

$high= count($a) - 1;

while($low <= $high){

$mid = intval(($low+$high)/2);

if($a[$mid] == $val) return $mid; //找到相等的就返回所在的位置

if($a[$mid] > $val){ //要查找的在前半截

$high = $mid - 1;

}else{

$low = $mid + 1; //要查找的在后半截

}

}

return -1;

}

//测试

$arr=array(1,3,6,8,10);

echo binarySearch($arr,3); //数组从0开始，故返回值是1

?>

1. **打印一个杨辉三角**

<?php

/\*\*=============================

\* 功能：生成杨辉三角数组。

\* 参数：$num 要生成的列数。

\*/

function generationYanghui($num){

$arr=array();

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

$arr[$i]=array();

//第一行和第二行均为1

if ($i==0){

$arr[$i][0]=1;

continue;

}

if ($i==1){

$arr[$i][0]=1;

$arr[$i][1]=1;

continue;

}

//从第三行开始，第一个和最后一个为1，

//第二个喂上一行的前一个位置的值和本位置的值的和。

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

if ($j==0){

$arr[$i][$j]=1;

continue;

}

if ($j==$i){

$arr[$i][$j]=1;

continue;

}

$arr[$i][$j]=$arr[$i-1][$j-1]+$arr[$i-1][$j];

}

}

return $arr;

}

//测试生成并打印出杨辉参数

$num=6;

$arrYH = generationYanghui($num);

for ($i=0;$i<sizeof($arrYH);$i++){

for($j=0;$j<sizeof($arrYH[$i]);$j++){

echo $arrYH[$i][$j] . " " ;

}

echo "<Br>";

}

?>

1. **使用PHP描述冒泡排序和快速排序算法，对象可以是一个数组**

冒泡排序算法：

<?php

//冒泡排序（一维数组）

function bubble\_sort($arr){

$count = count($arr);

if ($count <= 0) return false;

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

for($j=$count-1; $j>$i; $j--){

if ($arr[$j] < $arr[$j-1]){

$tmp = $arr[$j];

$arr[$j] = $arr[$j-1];

$arr[$j-1] = $tmp;

}

} }

return $arr;

}

//使用实例

$\_array = array('5', '8' ,'5' ,'6' ,'9' ,'3' ,'2' ,'4');

$\_array = bubble\_sort($\_array);

print\_r ($\_array);

?>

快速排序算法：

<?php

function quick\_sort($arr){

if(count($arr)<=1){

return $arr;

}

$key=$arr[0];

$left\_arr=array();

$right\_arr=array();

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

if($arr[$i]<=$key){

$left\_arr[]=$arr[$i];

}else{

$right\_arr[]=$arr[$i];

}

}

$left\_arr=quick\_sort($left\_arr);

$right\_arr=quick\_sort($right\_arr);

return array\_merge($left\_arr,array($key),$right\_arr);

}

//使用实例

$\_array = array('5', '8' ,'5' ,'6' ,'9' ,'3' ,'2' ,'4');

$\_array = quick\_sort($\_array);

print\_r ($\_array);

?>

1. **写一个二维数组排序函数，能够具有通用性**

/\*\*

\* 参数1 array 待排序的苏组

\* 参数2 string 第一项排序关键字

\* 参数3 string 排序方法，升序还是降序("SORT\_ASC"|"SORT\_DESC")

\* 参数4 string 排序类型，值有("SORT\_REGULAR"|"SORT\_NUMERIC"|"SORT\_STRING")

\* 返回值 array 排序后的数组

\*/

function MultiArraySort($ArrayData,$KeyName1,$SortOrder1 = "SORT\_ASC",$SortType1 = "SORT\_REGULAR")

{

if(!is\_array($ArrayData))

{

return $ArrayData;

}

// Get args number.

$ArgCount = func\_num\_args();

// Get keys to sort by and put them to SortRule array.

for($I = 1;$I < $ArgCount;$I ++)

{

$Arg = func\_get\_arg($I);

if(!eregi("SORT",$Arg))

{

$KeyNameList[] = $Arg;

$SortRule[] = '$'.$Arg;

}

else

{

$SortRule[] = $Arg;

}

}

// Get the values according to the keys and put them to array.

foreach($ArrayData AS $Key => $Info)

{

foreach($KeyNameList AS $KeyName)

{

${$KeyName}[$Key] = $Info[$KeyName];

}

}

// Create the eval string and eval it.

$EvalString = 'array\_multisort('.join(",",$SortRule).',$ArrayData);';

eval ($EvalString);

return $ArrayData;

}

1. **写一个function实现数组内每个元素之间加法的递代运算? 注:数组元素全部为自然数.**

**例:**

**$data=array(1,2,3);**

**程序要求打印出数组每个数组元素相互之间各种可能性加法(包括元素自已本身加法)**

1+1=2,1+2=3,1+3=5,

2+2=4,2+3=5,

3+3=6

打印出以下结果

Array

(

[0] => Array

(

[0] => 2

[1] => 3

[2] => 4

)

[1] => Array

(

[1] => 4

[2] => 5

)

[2] => Array

(

[2] => 6

)

)

<?php

function addArr($data){

$arr=array();

for ($i=0;$i<sizeof($data);$i++){

$arr[$i][0]=$data[$i]+$data[$i];

for($j=$i+1;$j<sizeof($data);$j++){

$arr[$i][$j]=$data[$i]+$data[$j];

}

}

print\_r($arr);

}

$data=array(1,2,3);

addArr($data);

?>

1. **设有n 种物品，每种物品有一个重量及一个价值。但每种物品的数量是无限的，同时有一个背包，最大载重量为M，今从n 种物品中选取若干件(同一种物品可以多次选取)，使其重量的和小于等于M，而价值的和为最大。  
   （假设有一个能装入总体积为T的背包和n件体积分别为w1 , w2 , … , wn 的物品，能否从n件物品中挑选若干件恰好装满背包，即使w1 +w2 + … + wn=T，要求找出所有满足上述条件的解。例如：当T=10，各件物品的体积{1，8，4，3，5，2}时，可找到下列4组解：（1，4，3，2） （1，4，5） （8，2） （3，5，2）。问题补充：提示：可利用回溯法的设计思想来解决背包问题。首先将物品排成一列，然后顺序选取物品装入背包，假设已选取了前i 件物品之后背包还没有装满，则继续选取第i+1件物品，若该件物品“太大”不能装入，则弃之而继续选取下一件，直至背包装满为止。但如果在剩余的物品中找不到合适的物品以填满背包，则说明“刚刚”装入背包的那件物品“不合适”，应将它取出“弃之一边”，继续再从“它之后”的物品中选取，如此重复，，直至求得满足条件的解，或者无解。 由于回溯求解的规则规则是“后进先出”因此自然要用到栈。 测试数据T=10 各物体体积{1，8，4，3，5，2} T=20 各物体体积{2，10，7，16，5，4，9，1}）**

<?php

/\*

设有n 种物品，每种物品有一个重量及一个价值。但每种物品的数量是无限的，同时有一个背包，最大载重量为M，今从n 种物品中选取若干件(同一种物品可以多次选取)，使其重量的和小于等于M，而价值的和为最大。

Input

第一行：两个整数，M(背包容量，M<= 200)和N(物品数量，N<= 30)； 第2..N+1 行：每行二个整数Wi,Ui，表示每个物品的重量和价值。

Output

仅一行，一个数，表示最大总价值。

Sample Input

12 4

2 1

3 3

4 5

7 9

Sample Output

15

\*/

set\_time\_limit(30);

define(’NOWTIME’,time());

function microtime\_float()

{

list($usec, $sec) = explode(" ", microtime());

$sec=$sec-NOWTIME;

return ((float)$usec + (float)$sec);

}

//初始化数组

/\*

$bbw=12;

$arywp=array();

$arywp[]=array(’d',2,1);

$arywp[]=array(’c',3,3);

$arywp[]=array(’b',4,5);

$arywp[]=array(’a',7,9);

\*/

$time\_start = microtime\_float();

//开始计算

$wpnum=count($arywp);

$wpnum=20; //允许的物品类型数量

$bbw=rand(5\*$wpnum,50\*$wpnum); //背包允许最大重量

for($i1=0;$i1<$wpnum;$i1++){ //初始化每个物品类型的重量和价值

$arywp[$i1][0]="w".$i1; //名称

$arywp[$i1][1]=rand(50,80); //重量

$arywp[$i1][2]=rand(20,50); //价值

$arywp[$i1][3]=$arywp[$i1][2]/$arywp[$i1][1];//单位重量的最大价值

}

function cmp($a, $b){

$returnvalue=0;

if ($a[3] == $b[3]) {

$returnvalue= 0;

} else {

$returnvalue = ($a[3] > $b[3]) ? -1 : 1;

}

return $returnvalue;

}

//按单位最大价值排序

usort($arywp, "cmp");

print\_r($arywp);

echo " <br />\n";

//递归函数，计算剩余空间的允许物品最大价值

function getmaxval($bbw,$arywp){

//最大单位物品的重量、价值、数量

$use\_w=0;

$use\_v=0;

$ary\_userwp=array();

//当前空间的允许物品最大价值

$maxwgt=0;

$maxval=0;

$maxuserwp=array();

$wp = array\_shift($arywp);

$num= floor(($bbw-$use\_w)/$wp[1]); //最大单位物品的最大允许个数

//echo " countmax : $num \* ".$wp[0]."<br/>\n";

if(count($arywp)==0 || $bbw-$use\_w==$wp[1]\*$num){ //没有剩余物品或刚好填满 //递归结束条件

if($num>0){

$maxwgt+=$wp[1]\*$num;

$maxval+=$wp[2]\*$num;

$maxuserwp[$wp[0]]=$num;

}

} else {

for($i=0;$i<=$num;$i++){ //计算最大单位物品不同数量时的最大价值

$use\_w=$wp[1]\*$i;

$use\_v=$wp[2]\*$i;

$ary\_userwp[$wp[0]]=$i;

if(($num-$i)\*$wp[2]/($bbw-$use\_w) < $arywp[0][3]){ //剩余空间的单位价值小于下件物品的单位价值

list($subuse\_w,$subuse\_v,$subusewp)=getmaxval($bbw-$use\_w,$arywp);

} else {

$subuse\_w=0;

$subuse\_v=0;

$subusewp=array();

}

if($use\_v+$subuse\_v>$maxval || $maxval==0){ //比较最大价值

$maxwgt=$use\_w+$subuse\_w;

$maxval=$use\_v+$subuse\_v;

if($i>0){

$maxuserwp=array\_merge($ary\_userwp,$subusewp);

} else{

$maxuserwp=$subusewp;

}

}

}

//print\_r($maxuserwp);

//echo " maxwgt : $maxwgt maxval : $maxval<br/>\n";

}

//返回 使用的重量，最大价值、使用的物品数量数组

return array($maxwgt,$maxval, $maxuserwp);

}

//开始计算

$arysulte= getmaxval($bbw,$arywp);

//输出结果

echo " bbw ".$bbw." ";

print\_r($arysulte);

$time\_end = microtime\_float();

$runtime = $time\_end – $time\_start;

echo " runtime : ".$runtime;

?>

1. **设随机抽到A的概率为0.1，B的概率为0.2，C的概率为0.3，D的概率为0.4，现在求按此概率随机抽出一个字母的算法**

<?php

/\*\*

\* 根据概率取随机数的算法（仅抽取1个）

\* 用法：

\* $proArr = array(10,20,30,40);

\* $result = pro\_rand($proArr);

\* echo '你抽到的$proArr数组索引是'. $result. '，其预设概率数是'. $proArr[$result];

\*\* @param array $proArr 概率数组。格式为array('A'=>10, 'B'=>40, 'C'=>50)，或者array(10,40,50)。数组的每个键值（value）必须为大于1的整数；所有数组键值（value）加起来即为其总概率精度

\* @return mixed $result 结果，将返回抽取到的概率数组索引值。

\*/

function pro\_rand( $proArr ){

$result = '';

//概率数组的总概率精度

$proSum = array\_sum($proArr);

foreach ( $proArr as $key => $proCur ){

$randNum = mt\_rand(1, $proSum);

if( $randNum <= $proCur ){

$result = $key;

break;

}else{

$proSum -= $proCur;

}

}

return $result;

}

?>

1. **一个小孩买了价值少于1美元的糖，并将1美元的钱交给售货员。售货员希望用数目最少的硬币个数找给小孩。假设提供了数目不限的面值为2 5美分、1 0美分、5美分、及1美分的硬币，写一个算法让售货员用最少的硬币个数找给小孩**
2. **五只猴子采得一堆桃子，猴子彼此约定隔天早起后再分食。不过，就在半夜里，一只猴子偷偷起来，把桃子均分成五堆后，发现还多一个，它吃掉这桃子，并拿走了其中一堆。第二只猴子醒来，又把桃子均分成五堆后，还是多了一个，它也吃掉这个桃子，并拿走了其中一堆。第三只，第四只，第五只猴子都依次如此分食桃子。那么桃子数最少应该有几个呢？**
3. **公鸡3元每只，母鸡5元每只，小鸡1元3只，一百元钱买一百只鸡。请求出公鸡，母鸡和小鸡的数目。**

<?php

//11：公鸡3元每只，母鸡5元每只，小鸡1元3只，一百元钱买一百只鸡。请求出公鸡，母鸡和小鸡的数目。

$totalmoney=100;

$totalchicken=100;

$cocknum = intval($totalmoney/3); //100元全买公鸡的数目

$hennum = intval($totalmoney/5); //100元全买母鸡的数目

$biddynum = 100; //100元虽然能买300只小鸡，但因为题意是要买100只鸡。所以小鸡数最大为100

for ($i=1;$i<$cocknum;$i++){ //公鸡数

for ($j=1;$j<$hennum;$j++){//母鸡数

for ($k=1;$k<$biddynum;$k++){//小鸡数

$money = $i \* 3 + $j \* 5 + $k / 3;

$total = $i + $j + $k;

if (($money == $totalmoney) &&($total == $totalchicken)){

echo "公鸡：" . $i . "只，母鸡：" . $j ."只，小鸡：" . $k ."只。<Br>";

}

}

}

}

?>

1. **小猴第一天摘下若干枣子，当即吃掉了一半，不过瘾又多吃了一个；第二天吃了剩下的一半又多吃了一个；以后每一天都吃了前一天剩下的一半多一个。到第十天小猴再想吃时，见到只剩下一只枣子了。问第一天这堆枣子有多少？**

<?php

//采用倒推算法。

function getTotal($day){

$total=0;

if ($day==1){ //第一天只有一个

$total = 1;

}else{ //以后每一天为前一天的两加上1个。

$total = 2\*(getTotal($day-1)+1);

}

return $total;

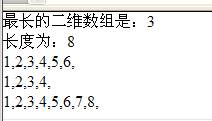
}

//测试

echo getTotal(10);

?>

1. **有一艘大船准备用来装载货物。所有待装货物都装在货箱中且所有货箱的大小都一样，但货箱的重量都各不相同。设第i 个货箱的重量为wi（1≤i≤n），而货船的最大载重量为c，如何在货船上装入最多的货物。**
2. **有一艘大船准备用来装载货物。所有待装货物都装在货箱中且所有货箱的大小都一样，但货箱的重量都各不相同。设第i 个货箱的重量为wi（1≤i≤n），而货船的最大载重量为c，如何在货船上装入最多的货物。**
3. **JS下有一个字符串var str = '1|2|3|4|5|6#1|2|3|4#1|2|3|4|5|6|7|8'; 请用“#”号将其分割为数组，再用“|”将其分割为2维数组。可以用例如ary[0][0]这样的各式取出数组中的每个元素。最后输出其中最长数组的编号(1,2,3这样的自然数)，这个数组中元素的个数。以及大数组中每个元素。如图：**



<script language=javascript>

var str = '1|2|3|4|5|6#1|2|3|4#1|2|3|4|5|6|7|8';

var arr=str.split("#"); //分割一维数组

var arrtwo=new Array();

for(var i=0;i<arr.length;i++){

arrtwo[i]=arr[i].split("|"); //分割二维数组。

}

var maxlength=arrtwo[0].length;

var maxarr=1;

//求二维数组最大长度和个数

for (var i=1;i<arrtwo.length;i++){

if (maxlength < arrtwo[i].length){

maxlength = arrtwo[i].length;

maxarr=i+1;

}

}

document.write("最长的二维数组是："+maxarr);

document.write("<br>长度为：" + maxlength);

//打印数组

for (var i=0;i<arrtwo.length;i++){

document.write("<br>");

for (var j = 0;j<arrtwo[i].length;j++){

document.write(arrtwo[i][j]+",");

}

}

</script>

1. **假设数据库中有两个表table1和table2，这两个表结构相同，都有三个字段id、name和description，分别存放编号、名称和描述。**

**现在请做一个页面，包含两个下拉菜单，第一个下拉菜单内是table1中的name，第二个下拉菜单内是table2中的name。下拉菜单的默认值是无，当选择了table1中的任意一个name时，出现一个textarea框显示name对应的description。当选择了table2中的任意一个name时，出现一个textarea框显示name对应的description，并且保留上一个下拉菜单的值和textarea中的值。Textarea中的值可修改，需要做到修改第一个textarea的值，然后更改第二个下拉菜单的值后，第一个textarea中的修改值仍然保留。最后是一个提交按钮，只有按下提交按钮，才会把textarea中的数据才会存入数据库。如图：**





**修改第一个菜单的中描述的值，然后选择第二个菜单的name值**



**提交后才会存入数据库：**



1. **有一个数组array('a','b','c','d','e','f')，请使用两段代码分别达成图1和图2中的效果。**

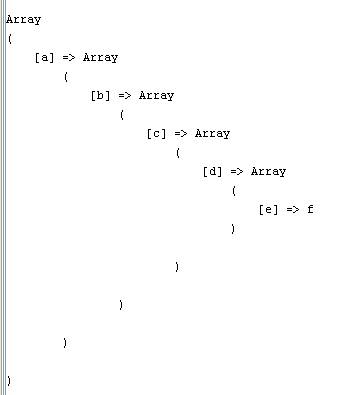
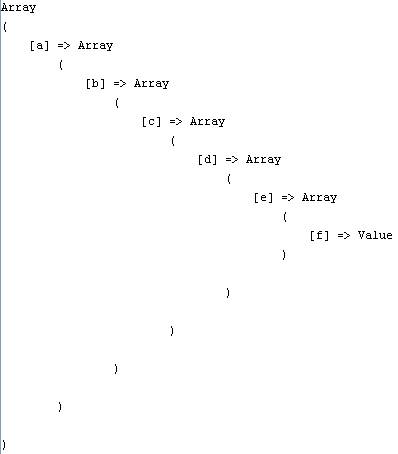
图1：

图2：

第一种形式：

<?php

/\*\*

\* 产生第一种形式数组

\* 参数：数组

\* 返回：数组

\*/

function generateArray1($arr){

if (sizeof($arr)<2) return $arr; //如果数组个数只有一个或者0个，返回原始值。

$r\_arr=null;

for ($i=sizeof($arr) - 2;$i>=0;$i--){

if ($i == sizeof($arr) - 2){

$r\_arr=array($arr[$i]=>$arr[sizeof($arr)-1]);

}else{

$r\_arr=array($arr[$i]=>$r\_arr);

}

}

return $r\_arr;

}

$test\_arr=array('a','b','c','d','e','f');

print\_r(generateArray1($test\_arr));

?>

第二种形式：

<?php

/\*\*

\* 产生第二种形式数组

\* 参数：数组

\* 传入的值

\* 返回：数组

\*/

function generateArray2($arr,$value){

if (sizeof($arr)<1) return $arr; //如果数组个数只有0个，返回原始值。

$r\_arr=null;

for ($i=sizeof($arr) - 1;$i>=0;$i--){

if ($i == sizeof($arr) - 1){

$r\_arr=array($arr[$i]=>$value);

}else{

$r\_arr=array($arr[$i]=>$r\_arr);

}

}

return $r\_arr;

}

$test\_arr=array('a','b','c','d','e','f');

print\_r(generateArray2($test\_arr,'Value'));

?>

1. **请使用一句话将类似“1910-09-09”的日期各式转换为“一九一零年九月九日”这样的各式。(需要有通用性，是不是一句话不是关键，但不允许使用if\switch\while\for等语句)。**

<?php

function convertNumberToChinese($str){

//先将-替换成年月日

$pattern='/(\d+)-(0\*)([1-9]+)-(0\*)([1-9]+)/';

$replace='${1}年${3}月${5}日';

$str= preg\_replace($pattern,$replace,$str);

//再将数字替换成汉字

$patterns=array("/0/","/1/","/2/","/3/","/4/","/5/","/6/","/7/","/8/","/9/");

$replaces=array("零","一","二","三","四","五","六","七","八","九");

$re = preg\_replace($patterns,$replaces,$str);

return $re;

}

$str = '1910-09-09';

echo convertNumberToChinese($str)

?>

1. **有一张表test，包含四个字段id, question, answer，rank里面有1000条数据。要求做一个页面，随机抽取50道题目（都是单选题），分页显示所有的题目，每页显示1题。按钮包含：第一题—上一页—当前第N题—下一页—最后一题。交卷后将每一题的答案和数据库中的标准答案作比较，相同+2分，最后得出成绩存入rank。**
2. **简述如何得到当前执行脚本路径，包括所得到参数。**

**说明：例如有一个脚本www.domain.com,传给他的参数有参数1，参数2，参数3….**

**传递参数的方法有可能是GET有可能是POST,那么现在请写出类似**

**http://www.domain.com/script.php? 参数1=值1&参数2=值2..... 的结果**

PHP代码如下，比如文件名是allparam.php

<?php

$query\_string = "";

if ($\_POST) { //如果是post方法，取得全部post参数及值后进行连接处理

$kv = array();

foreach ($\_POST as $key => $value) {

$kv[] = "$key=$value";

}

$query\_string = join("&", $kv);

}

else { //非post方法，GET等。

$query\_string = $\_SERVER['QUERY\_STRING'];

}

$execfile=$\_SERVER["REQUEST\_URI"];

if (strstr($execfile,"?")){ //如果请求URI中包含?则执行文件取?前面的部分

$execfile = substr($execfile,0,strlen($execfile) - strlen(strstr($execfile,"?")));

}

echo "http://".$\_SERVER["SERVER\_NAME"].$execfile."?".$query\_string;

?>

测试代码，post测试(这个html文件和allparam.php文件在同一个目录里)

<html>

<head>

<title>Param Test </title>

</head>

<body>

<form action="allparam.php" method="post">

Name:<input type=text value="" name="name"><br>

Addr:<input type=text value="" name="address"><br>

<input type=submit value="提交">

</form>

</body>

</html>

get测试：

<http://localhost/allparam.php?c=dd&e=xx>

1. **写出一个能创建多级目录的PHP函数**

<?php

function createdir($path,$mode){

if (is\_dir($path)){ //判断目录存在否，存在不创建

echo "目录'" . $path . "'已经存在";

}else{ //不存在创建

$re=mkdir($path,$mode,true); //第三个参数为true即可以创建多极目录

if ($re){

echo "目录创建成功";

}else{

echo "目录创建失败";

}

}

}

$path="/aa/bb/cc/cd"; //要创建的目录

$mode=0755; //创建目录的模式

createdir($path,$mode);//测试

?>

1. **有一个留言板，用mysql做数据库**

**用户信息包括：用户名，密码，email**

**留言内容包括：留言ID，标题，内容，发表时间，状态（审核，未审核）**

**请实现下列需求：**

**1）数据库结构。无需写建表语句，用类似下面的表格，描述清楚即可**

**表明 table\_aaa**

**字段名 字段说明 字段类型 索引**

**name 姓名 Varchar（64） 唯一索引**

**gender 性别 Enum（‘M’，‘F’）**

**注意，要在索引栏中注明是否需要创建索引，以及要创建的索引的类型**

**用一个sql语句查询出发表留言数量大于10条的用户名及其留言数量，查询结果按留言数量降序排列**

用户信息表：

表名：tbl\_user

username 用户名 varchar(20) 唯一索引

passwd1 密码 varchar(20)

email email varchar(100)

留言内容表：

表名：tbl\_content

conid 留言ID int

title 标题 varchar(100)

content 内容 varchar(4000)

devtime 发表时间 datetime

state 状态（审核，未审核） varchar(1)

username 用户名 varchar(20) 可重复索引

select username,count(\*)

from tbl\_content

group by username

having count(\*) > 10

order by count(\*) desc

1. **写出php的public、protected、private三种访问控制模式的区别**

public是所有的类中都能访问，protected是只能在本类及继承该类的子类能访问。private只能在本类中访问。

1. **给你三个数，写程序求出其最大值。**

<?php

function getMax($a,$b,$c){

$max = $a;

if ($b > $max){

$max = $b;

}

if ($c > $max){

$max = $c;

}

return $max;

}

echo getMax(3,2,15);

?>

1. **写出查询发贴数最多的十个人名字的SQL，利用下表：members(id,username,posts,pass,email)**

我理解posts是发帖量（整型）。所以我的SQL语句如下：

select username from members order by posts desc limit 10;

1. **如何通过javascript判断一个窗口是否已经被屏蔽。**

<script type="text/javascript" language="javascript">

/\*\*

\* 参数说明：url - 要打开的窗口

\* mode - true 模态打开 false 非模态打开

\* 返回值：boolean true 没有屏蔽 false 被屏蔽

\*/

function checkshield(url,mode){

var breturn = true;

if (mode){ //模态打开

if (window.open(url) == false) return false;

}else{ //非模态打开

try{

window.showDialogMode(url);

}catch (err){

breturn = false;

}

}

return breturn;

}

</script>

1. **有一数组　$a=array(4,3,8,9,2);请将其重新排序（按从小到大的顺序列出）。**

<?php

$a=array(4,3,8,9,2);

for($i=0;$i<sizeof($a);$i++){

for ($j = $i + 1;$j<sizeof($a);$j++){

if ($a[$i] > $a[$j]){

$itemp = $a[$i];

$a[$i]=$a[$j];

$a[$j] = $itemp;

}

}

}

print\_r($a);

?>

1. **写出匹配邮箱地址和URL的两个正则表达式。类似下面的：**

**邮箱地址：**[**user\_name.first@hztraining.com**](mailto:user_name.first@hztraining.com)

**URL地址：**[**http://www.hztraining.com/user\_profile.php?uid=100**](http://www.hztraining.com/user_profile.php?uid=100)

<?php

//匹配Email

$pattern='/(\w+)\@(\w+)\.((\w+))+/';

$email='wangchc\_chc.xx@163.xx';

if (preg\_match($pattern,$email))

echo "match";

else

echo "not match";

echo "<Br>";

//匹配Url

$patternurl='/http(s?):\/\/(((\w+)\.)+)(\w+)\/(\w-\.\?\%\&\=)\*/';

$url="http://www.hztraining.com/user\_profile.php?uid=100";

if (preg\_match($patternurl,$url))

echo "match";

else

echo "not match";

?>

1. **对于大流量的网站,您采用什么样的方法来解决访问量问题?**

首先，确认服务器硬件是否足够支撑当前的流量。

其次，优化数据库访问。

第三，禁止外部的盗链。

第四，把持大文件的下载。

第五，应用不同主机分流重要流量

第六，应用流量分析统计软件。

1. **MySQL数据库，一天一万条以上的增量，怎么优化？**

首先应该做好mysql基本优化,字段大小,索引;

另外根据具体的需求看有没有必要建立二个一样的表,一个是存储历史数据,一个是当天的或者最近多少天的;

根据日期建立索引这个应该有必要;

1. **写个函数来解决多线程同时读写一个文件的问题。**

<?php

/\*\*

\* 参数说明：$filename string 操作文件名

\* $content string 写入文件内容，读时传入空就可以

\* $mode string r-读 w-写

\* 返回值 $retstr string 返回信息(读取时返回文件内容)

\*/

function multiAccessFile($filename,$content,$mode){

$retstr = "";

if (file\_exists($filename) == false) {

$retstr = "文件不存在";

}

if ($mode == "w"){

$file=fopen($filename,"w+");

if(flock($file,LOCK\_EX))

{

fwrite($file,$content);

flock($file,LOCK\_UN);

$retstr="文件写入成功";

}

else{

$retstr = "文件写入时发生错误";

}

}

if ($mode == "r"){

$file = fopen($filename,"r");

if (flock($file,LOCK\_EX))

{

$retstr = fread($file,filesize($filename));

flock($file,LOCK\_UN);

}else{

$retstr = "读取文件时发生错误";

}

}

fclose($file);

return $retstr;

}

?>

1. **一个一维数组，里面存储整形数据，请写一个函数，将他们按从大到小的顺序排列。要求执行效率高。并说明如何改善执行效率。（该函数必须自己实现，不能使用php函数）**

<?php

/\*\*

\*功能：32、一个一维数组，里面存储整形数据，请写一个函数，将他们按从大到小的顺序排列。要求执行效率高。并说明如何改善执行效率。（该函数必须自己实现，不能使用php函数）

\*参数：待排序数组

\*返回：排序后数组

\*/

function sortArr($arr){

if (sizeof($arr)<2) return $arr; //数组元素为1个或者0个，返回。

/\*

\*借鉴冒泡算法，但又和算法不一样。执行次数应该为(n-1)+(n-2)+.....+1次，其他的优化方法暂时没有思路

\*/

for ($i=0;$i<sizeof($arr);$i++){

for ($j=$i+1;$j<sizeof($arr);$j++){

if ($arr[$i]<$arr[$j]){ //用第一个和后面的比较，保证第一个最大；以此类推，第二个和以后的比较

$temp = $arr[$i];

$arr[$i] = $arr[$j];

$arr[$j] = $temp;

}

}

}

return $arr;

}

//测试

$test\_arr=array(10,2,5,1,9,21,13);

print\_r(sortArr($test\_arr));

?>

1. **写一个函数，算出两个文件的相对路径**

如 $a = '/a/b/c/d/e.php';

　　$b = '/a/b/12/34/c.php';

计算出 $b 相对于 $a 的相对路径应该是 ../../c/d将()添上

<?php

$a = '/c/d/d/e.php';

$b = '/a/c/d/e.php';

$aArr = explode('/',dirname($a));

$bArr = explode('/',dirname($b));

$aLen = count($aArr);

$bLen = count($bArr);

$maxLen = max($aLen,$bLen);

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

if($aArr[$i] != $bArr[$i] && isset($aArr[$i])){

if(isset($bArr[$i]))$bUrl[]=$bArr[$i];

$relativePath .= "../";

}else{

$bUrl[]=$bArr[$i];

}

}

echo $relativePath.implode('/',$bUrl).'/'.basename($b);

?>

1. **用写一个类实现会员管理，要求实现：**

1）用文件存储会员信息，会员注册输入用户名和电子邮件就行。

2）用户信息包括：用户名，电子邮件。

3）要求用户可以登录、退出和注销用户。

4）如果用户没有退出，下次登录自动显示用户名。

5）保存用户上次浏览时间。