1 - 编程能力检验

时间要求

1-2周

任务

请不要调用任何系统的函数，用python 或 Scala 最基础语法实现以下函数功能

数组

array\_change\_key\_case — 将数组中的所有键名修改为全大写或小写

array\_chunk — 将一个数组分割成多个

array\_column — 返回数组中指定的一列

array\_combine — 创建一个数组，用一个数组的值作为其键名，另一个数组的值作为其值

array\_count\_values — 统计数组中所有的值

array\_diff\_assoc — 带索引检查计算数组的差集

array\_diff\_key — 使用键名比较计算数组的差集

array\_diff\_uassoc — 用用户提供的回调函数做索引检查来计算数组的差集

array\_diff\_ukey — 用回调函数对键名比较计算数组的差集

array\_diff — 计算数组的差集

array\_fill\_keys — 使用指定的键和值填充数组

array\_fill — 用给定的值填充数组

array\_filter — 用回调函数过滤数组中的单元

array\_flip — 交换数组中的键和值

array\_intersect\_assoc — 带索引检查计算数组的交集

array\_intersect\_key — 使用键名比较计算数组的交集

array\_intersect\_uassoc — 带索引检查计算数组的交集，用回调函数比较索引

array\_intersect\_ukey — 用回调函数比较键名来计算数组的交集

array\_intersect — 计算数组的交集

array\_key\_exists — 检查数组里是否有指定的键名或索引

array\_key\_first — Gets the first key of an array

array\_key\_last — Gets the last key of an array

array\_keys — 返回数组中部分的或所有的键名

array\_map — 为数组的每个元素应用回调函数

array\_merge\_recursive — 递归地合并一个或多个数组

array\_merge — 合并一个或多个数组

array\_multisort — 对多个数组或多维数组进行排序

array\_pad — 以指定长度将一个值填充进数组

array\_pop — 弹出数组最后一个单元（出栈）

array\_product — 计算数组中所有值的乘积

array\_push — 将一个或多个单元压入数组的末尾（入栈）

array\_rand — 从数组中随机取出一个或多个单元

array\_reduce — 用回调函数迭代地将数组简化为单一的值

array\_replace\_recursive — 使用传递的数组递归替换第一个数组的元素

array\_replace — 使用传递的数组替换第一个数组的元素

array\_reverse — 返回单元顺序相反的数组

array\_search — 在数组中搜索给定的值，如果成功则返回首个相应的键名

array\_shift — 将数组开头的单元移出数组

array\_slice — 从数组中取出一段

array\_splice — 去掉数组中的某一部分并用其它值取代

array\_sum — 对数组中所有值求和

array\_udiff\_assoc — 带索引检查计算数组的差集，用回调函数比较数据

array\_udiff\_uassoc — 带索引检查计算数组的差集，用回调函数比较数据和索引

array\_udiff — 用回调函数比较数据来计算数组的差集

array\_uintersect\_assoc — 带索引检查计算数组的交集，用回调函数比较数据

array\_uintersect\_uassoc — 带索引检查计算数组的交集，用单独的回调函数比较数据和索引

array\_uintersect — 计算数组的交集，用回调函数比较数据

array\_unique — 移除数组中重复的值

array\_unshift — 在数组开头插入一个或多个单元

array\_values — 返回数组中所有的值

array\_walk\_recursive — 对数组中的每个成员递归地应用用户函数

array\_walk — 使用用户自定义函数对数组中的每个元素做回调处理

array — 新建一个数组

arsort — 对数组进行逆向排序并保持索引关系

asort — 对数组进行排序并保持索引关系

compact — 建立一个数组，包括变量名和它们的值

count — 计算数组中的单元数目，或对象中的属性个数

current — 返回数组中的当前单元

each — 返回数组中当前的键／值对并将数组指针向前移动一步

end — 将数组的内部指针指向最后一个单元

extract — 从数组中将变量导入到当前的符号表

in\_array — 检查数组中是否存在某个值

key\_exists — 别名 array\_key\_exists

key — 从关联数组中取得键名

krsort — 对数组按照键名逆向排序

ksort — 对数组按照键名排序

list — 把数组中的值赋给一组变量

natcasesort — 用“自然排序”算法对数组进行不区分大小写字母的排序

natsort — 用“自然排序”算法对数组排序

next — 将数组中的内部指针向前移动一位

pos — current 的别名

prev — 将数组的内部指针倒回一位

range — 根据范围创建数组，包含指定的元素

reset — 将数组的内部指针指向第一个单元

rsort — 对数组逆向排序

shuffle — 打乱数组

sizeof — count 的别名

sort — 对数组排序

uasort — 使用用户自定义的比较函数对数组中的值进行排序并保持索引关联

uksort — 使用用户自定义的比较函数对数组中的键名进行排序

usort — 使用用户自定义的比较函数对数组中的值进行排序

字符串

addcslashes — 以 C 语言风格使用反斜线转义字符串中的字符

addslashes — 使用反斜线引用字符串

bin2hex — 函数把包含数据的二进制字符串转换为十六进制值

chop — rtrim 的别名

chr — 返回指定的字符

chunk\_split — 将字符串分割成小块

convert\_cyr\_string — 将字符由一种 Cyrillic 字符转换成另一种

convert\_uudecode — 解码一个 uuencode 编码的字符串

convert\_uuencode — 使用 uuencode 编码一个字符串

count\_chars — 返回字符串所用字符的信息

crc32 — 计算一个字符串的 crc32 多项式

crypt — 单向字符串散列

echo — 输出一个或多个字符串

explode — 使用一个字符串分割另一个字符串

fprintf — 将格式化后的字符串写入到流

get\_html\_translation\_table — 返回使用 htmlspecialchars 和 htmlentities 后的转换表

hebrev — 将逻辑顺序希伯来文（logical-Hebrew）转换为视觉顺序希伯来文（visual-Hebrew）

hebrevc — 将逻辑顺序希伯来文（logical-Hebrew）转换为视觉顺序希伯来文（visual-Hebrew），并且转换换行符

hex2bin — 转换十六进制字符串为二进制字符串

html\_entity\_decode — Convert HTML entities to their corresponding characters

htmlentities — 将字符转换为 HTML 转义字符

htmlspecialchars\_decode — 将特殊的 HTML 实体转换回普通字符

htmlspecialchars — 将特殊字符转换为 HTML 实体

implode — 将一个一维数组的值转化为字符串

join — 别名 implode

lcfirst — 使一个字符串的第一个字符小写

levenshtein — 计算两个字符串之间的编辑距离

localeconv — Get numeric formatting information

ltrim — 删除字符串开头的空白字符（或其他字符）

md5\_file — 计算指定文件的 MD5 散列值

md5 — 计算字符串的 MD5 散列值

metaphone — Calculate the metaphone key of a string

money\_format — 将数字格式化成货币字符串

nl\_langinfo — Query language and locale information

nl2br — 在字符串所有新行之前插入 HTML 换行标记

number\_format — 以千位分隔符方式格式化一个数字

ord — 返回字符的 ASCII 码值

parse\_str — 将字符串解析成多个变量

print — 输出字符串

printf — 输出格式化字符串

quoted\_printable\_decode — 将 quoted-printable 字符串转换为 8-bit 字符串

quoted\_printable\_encode — 将 8-bit 字符串转换成 quoted-printable 字符串

quotemeta — 转义元字符集

rtrim — 删除字符串末端的空白字符（或者其他字符）

setlocale — 设置地区信息

sha1\_file — 计算文件的 sha1 散列值

sha1 — 计算字符串的 sha1 散列值

similar\_text — 计算两个字符串的相似度

soundex — Calculate the soundex key of a string

sprintf — Return a formatted string

sscanf — 根据指定格式解析输入的字符

str\_getcsv — 解析 CSV 字符串为一个数组

str\_ireplace — str\_replace 的忽略大小写版本

str\_pad — 使用另一个字符串填充字符串为指定长度

str\_repeat — 重复一个字符串

str\_replace — 子字符串替换

str\_rot13 — 对字符串执行 ROT13 转换

str\_shuffle — 随机打乱一个字符串

str\_split — 将字符串转换为数组

str\_word\_count — 返回字符串中单词的使用情况

strcasecmp — 二进制安全比较字符串（不区分大小写）

strchr — 别名 strstr

strcmp — 二进制安全字符串比较

strcoll — 基于区域设置的字符串比较

strcspn — 获取不匹配遮罩的起始子字符串的长度

strip\_tags — 从字符串中去除 HTML 和 PHP 标记

stripcslashes — 反引用一个使用 addcslashes 转义的字符串

stripos — 查找字符串首次出现的位置（不区分大小写）

stripslashes — 反引用一个引用字符串

stristr — strstr 函数的忽略大小写版本

strlen — 获取字符串长度

strnatcasecmp — 使用“自然顺序”算法比较字符串（不区分大小写）

strnatcmp — 使用自然排序算法比较字符串

strncasecmp — 二进制安全比较字符串开头的若干个字符（不区分大小写）

strncmp — 二进制安全比较字符串开头的若干个字符

strpbrk — 在字符串中查找一组字符的任何一个字符

strpos — 查找字符串首次出现的位置

strrchr — 查找指定字符在字符串中的最后一次出现

strrev — 反转字符串

strripos — 计算指定字符串在目标字符串中最后一次出现的位置（不区分大小写）

strrpos — 计算指定字符串在目标字符串中最后一次出现的位置

strspn — 计算字符串中全部字符都存在于指定字符集合中的第一段子串的长度。

strstr — 查找字符串的首次出现

strtok — 标记分割字符串

strtolower — 将字符串转化为小写

strtoupper — 将字符串转化为大写

strtr — 转换指定字符

substr\_compare — 二进制安全比较字符串（从偏移位置比较指定长度）

substr\_count — 计算字串出现的次数

substr\_replace — 替换字符串的子串

substr — 返回字符串的子串

trim — 去除字符串首尾处的空白字符（或者其他字符）

ucfirst — 将字符串的首字母转换为大写

ucwords — 将字符串中每个单词的首字母转换为大写

vfprintf — 将格式化字符串写入流

vprintf — 输出格式化字符串

vsprintf — 返回格式化字符串

wordwrap — 打断字符串为指定数量的字串

学习笔记

请为每个实现的系统函数，写一篇blog（含思路，开发过程）