

C语言实验六：字符串应用和结构体定义

必做题要求 (勿忘提交截图, 尤其是做了选做题的同学)

1. 课程内完成: OJ系统的U~AB题, 和AJ题
2. 提交要求: 提交包含以上题目编号的截图, 截图右上角必须包含你的学号信息

NWAUOJ							2020110020(孙海天)
	U	C语言实习题四--6.姓名排序	274	606			
	V	C语言实习题四--7.输出杨辉三角形	292	619			
	W	C语言实习题五--1.用指针实现排序	278	495			
	X	C语言实习题五--2.用指针实现字符串排序	230	462			
	Y	C语言实习题五--3.数据倒置	257	377			
	Z	C语言实习题五--4.用指针实现数据位置调整	226	737			
	AA	C语言实习题五--5.用指针实现查找二维数组中最大数及其位置	216	405			
	AB	C语言实习题五--6.用指针实现子字符串提取	183	398			
	AC	C语言实习题六--1.整数交换函数设计	252	360			
	AD	C语言实习题六--2.数字字符串个数统计函数设计	211	415			
	AE	C语言实习题六--3.排序函数设计	219	377			
	AF	C语言实习题六--4.矩阵转置函数设计	198	379			
	AG	C语言实习题六--5.求素数函数设计	218	546			
	AH	C语言实习题六--7.求最大公约数函数设计	223	380			
	AI	C语言实习题六--8.对称数组判断	170	392			
	AJ	C语言实习题七--1.结构体的定义与引用	122	255			

选做题要求

- 按以下要求提交, 否则不计分。需提交:
 1. 相应函数实现的截图;
 2. 用相应测试例测试的运行结果截图。
- 若题目要求使用动态数组实现却未合理释放内存, 则该题不计分。
- 完成奖励:
 - 全部做完并检查通过后则可以提前离开课堂
 - 晚上9:15前完成并提交: 平时成绩总分每题加0.1分 (加满为止, 只通过部分测试例的不加分)

题目1：动态字符串数组的双重排序

函数原型:

```
1 void SortStrings(char **strs, int n);
```

功能描述：

编写一个函数 `SortStrings`，对给定的字符串数组 `strs` 进行如下处理：

1. 对每个字符串内部的字符按 ASCII 码升序排列（即每个字符串内字符重新排序）；
2. 再将所有字符串按字典序升序排列。

示例：

- 输入： {"cba", "fed", "abc"}
- 第一步后： {"abc", "def", "abc"}
- 第二步后： {"abc", "abc", "def"}

参数说明：

- `strs`：指向字符串数组的指针，每个元素是一个 `char*` 类型的字符串
- `n`：字符串的数量

返回值：

- 无返回值，直接修改原数组内容

测试代码下载方式：

```
1 wget http://10.131.21.39:8000/TestSortStrings.c
```

题目2：模拟解析CSV文件

函数原型：

```
1 void ParseCSV(const char *csvData, struct Person **persons, int *count);
```

功能描述： 将一个模拟的 CSV 格式字符串按行解析，每行包含三列：姓名 (`char name[20]`)、年龄 (`int age`)、城市 (`char city[20]`)。将每一行数据转换为一个 `struct Person` 结构体，并存储在动态分配的数组中。

参数说明：

- `csvData`：输入的 CSV 字符串，格式如 "Zoe,29,Hangzhou\nAlice,25,Beijing"
- `persons`：指向一维结构体数组的二级指针，用于接收解析后的 `Person` 结构体数组
- `count`：指向整数，返回解析出的记录总数

返回值：

- 无返回值

说明：

1. `csvData` 每行以 `\n` 分隔，字段（列）以 `,` 分隔；

2. 若 `csvData` 为空或 `NULL`, 则 `*count = 0, *persons = NULL` ;
3. 若检查到 `csvData` 有其他错误, 也使 `*count = 0, *persons = NULL` 。

测试代码下载方式:

```
1 wget http://10.131.21.39:8000/TestParseCSV.c
```

题目3：排序后存储为CSV风格的字符串

函数原型:

```
1 char* SortSaveAsCSV(struct Person *persons, int count);
```

功能描述:

将一个 `struct Person` 类型的结构体数组按年龄升序排序, 并转换为 CSV 格式的字符串。每行包含三列: 姓名 (`char name[20]`) 、年龄 (`int age`) 、城市 (`char city[20]`) , 字段 (列数据) 之间用逗号分隔, 行之间用换行符 `\n` 分隔。

最后一行末尾也必须包含换行符 `\n` 。

假设年龄最大为100岁, 最小为-10岁 (允许错误输入)

参数说明:

- `persons` : 指向结构体数组的指针, 存储待序列化的数据;
- `count` : 数组中有效元素的个数 (≥ 0) ;

返回值:

- 返回一个动态分配的字符串, 内容为 CSV 格式;
- 若 `count == 0` 或 `persons == NULL` , 返回空字符串 `""` (即长度为1的字符串, 仅含 `\0`) ;
- 调用者负责使用 `free()` 释放该内存。

实现要求:

- 每行格式为: `name,age,city\n` ;
- 所有字段直接转换, 不加引号;
- 字段中可能包含数字、下划线、连字符等特殊字符, 需原样输出 (无需转义) ;

测试代码下载方式:

```
1 wget http://10.131.21.39:8000/TestSortSaveAsCSV.c
```