

# C语言实验五：数组和字符串

## 必做题要求

1. 课程内完成：OJ系统的O~T题
2. 提交要求：提交如图所示的截图，截图右上角必须包含你的学号信息

The screenshot shows a competition list from NWAFU-OJ. The top navigation bar includes links for Home, Competition & Assignment List, Problem List, Status List, Ranking, OI-Ranking, and Statistics. The right side of the header displays the user information: 2020110020(孙海天) ▾.

| Y | O | C语言实习题三--5.图形输出      | 399 | 1110 |
|---|---|----------------------|-----|------|
| Y | P | C语言实习题四--1.字符串连接     | 296 | 566  |
| Y | Q | C语言实习题四--2.简单加密程序    | 248 | 653  |
| Y | R | C语言实习题四--3.矩阵对角线元素求和 | 262 | 580  |
| Y | S | C语言实习题四--4.数据顺序调整    | 232 | 1407 |
| Y | T | C语言实习题四--5.字符个数统计    | 226 | 706  |

## 选做题要求

- 共5题，除 `strlen`, `strcpy`, `strcat` 和 `strcmp` 外，禁止调用 `string.h` 中的其他函数。
- 提交时需标明题号，仅按以下要求提交，否则不计分。需提交：
  1. 相应函数实现的截图，或修改部分的相关说明；
  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：
  - 为避免交叉感染，本周不执行“提前检查通过则可以离开”
  - 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）

## 题目1：字符串查找函数

函数原型：

```
1 int FindSubstring(const char* str, const char* substr);
```

功能描述：

实现字符串查找功能，在字符串 `str` 中查找子串 `substr` 是否存在。

参数说明：

- `str`：主字符串（被查找的字符串）
- `substr`：要查找的子字符串

**返回值：**

- 如果 `substr` 存在于 `str` 中，返回1
- 如果 `substr` 不存在于 `str` 中，返回0

**测试代码下载方式：**

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

## 题目2：字符串替换函数

**函数原型：**

```
1 char* ReplaceAll(const char* original, const char* target, const char* replacement);
```

**功能描述：**

将字符串 `original` 中所有出现的子串 `target` 替换为字符串 `replacement`。

**参数说明：**

- `original`：原始字符串
- `target`：要被替换的子串
- `replacement`：替换后的字符串

**返回值：**

- 返回替换后的新字符串的指针
- 如果内存分配失败，返回NULL

**要求：**

- 测试代码的 `ReplaceAll` 实现中有多处错误，请下载分析并修改（建议分析时用debug查看）
- 注意：除 `ReplaceAll` 以外没有错误，请勿随意修改其他代码
- 同时注意符合内存泄漏检查的要求

**测试代码下载方式：**

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

## 题目3：回文单词检测函数

**函数原型：**

```
1 int HasPalindromeWord(const char* sentence);
```

**功能描述：**

检测句子中是否存在回文单词。回文单词是指正着读和反着读都一样的单词（忽略大小写和标点符号）。

**参数说明：**

- `sentence` : 输入的句子字符串

返回值：

- 如果句子中存在至少一个回文单词，返回1
- 如果句子中不存在回文单词，返回0

实现要求：

- 注意：单个字符的单词不属于回文单词
- 需要正确处理单词分割（空格和标点作为分隔符）
- 判断回文时忽略字母大小写和标点符号
- 若需要，可适当增加辅助函数（例如判断一个单词是否回文的函数）

测试代码下载方式：

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

## 题目4：礼貌数判断函数（被放弃的期中考试题）

函数原型：

```
1 int IsPoliteNumber(int n);
```

功能描述：

判断一个正整数是否为礼貌数。

某些正整数可以表示为两个或更多连续正整数之和，这种数字称为礼貌数（Polite Number）。

例如：

$9=4+5$ ,  $9=2+3+4$ , 因此9是礼貌数；

8无法表示为两个或更多连续正整数之和，因此8不是礼貌数。

注意：连续正整数的个数至少为2。

参数说明：

- `n` : 要判断的正整数

返回值：

- 如果 `n` 是礼貌数，返回1
- 如果 `n` 不是礼貌数，返回0

测试代码下载方式：

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

## 题目5：矩阵转置函数

函数原型：

```
1 int** TransMat(int** matrix, int m, int n);
```

### 功能描述:

实现矩阵的转置操作，将 $m \times n$ 的矩阵转置为 $n \times m$ 的矩阵。

矩阵转置的简单理解：行变列，列变行。第一行变成第一列，第二行变成第二列。

### 矩阵转置示例

原始矩阵 (3行2列)：

```
1 10 20  
2 30 40  
3 50 60
```

转置后矩阵 (2行3列)：

```
1 10 30 50  
2 20 40 60
```

### 参数说明:

- `matrix` : 原始矩阵 (动态二维数组)
- `m` : 原始矩阵的行数
- `n` : 原始矩阵的列数

### 返回值:

- 返回转置后的新矩阵 ( $n \times m$ 的二维数组)
- 如果内存分配失败，返回NULL

### 实现要求:

- 必须使用二维动态数组实现
- 调用者负责释放返回的矩阵内存

### 示例:

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