

# C 语言实训教程

## ----函数专项练习

### 一、实验目的

1. 熟练进行函数的声明、定义、调用；
2. 能够熟练运用函数解决实际问题；
3. 正确使用函数递归解决实际问题；
4. 能灵活运用常用系统函数解决问题。

### 二、实验内容及实验步骤

#### (一) 编程题

1. 编写一个 `void sort(int *x, int n)` 实现将 `x` 数组中的 `n` 个整数从大到小排序。  
`n` 及数组元素在主函数中输入，将结果显示在屏幕上。

2-1. 编写函数 `replace(char *s, char c1, char c2)` 实现将 `s` 所指向的字符串中所有字符 `c1` 用 `c2` 替换，字符串、字符 `c1` 和 `c2` 均在主函数中输入，将原始字符串和替换后的字符串显示在屏幕上。

2-2. 字符串替换

#### 习题描述：

编写函数 `replace(char *str, char *fstr, char *rstr)`，将 `str` 所指字符串中凡是与 `fstr` 字符串相同的字符替换成 `rstr`（`rstr` 与 `fstr` 的字符长度不一定相同）。从主函数中输入原始字符串 "ifordowhileelsewhilebreak"、查找字符串 "while" 和替换字符串 "struct"，调用函数得到结果。

#### 输入描述：

第1行输入要字符串 `str`，第2行输入查找字符串 `fstr`，第3行输入替换字符串 `rstr`。

#### 输入示例：

```
ifordowhileelsewhilebreak
while
struct
```

#### 输出示例：

```
ifordostructelsestructbreak
```

#### 实际应用：

百度语言翻译机

百度的工程师们是非常注重效率的，在长期的开发与测试过程中，他们逐渐创造了一套独特的缩略语。他们在平时的交谈、会议，甚至在各种技术文档中都会大量运用。

为了让新员工可以更快地适应百度的文化，更好地阅读公司的技术文档，人力资源部决定开发一套专用的翻译系统，把相关文档中的缩略语和专有名词翻译成日常语言。

输入要求：

输入数据包含三部分：

1. 第一行包含一个整数 `N` ( $N \leq 10000$ )，表示总共有多少个缩略语的词条；

2. 紧接着有 N 行的输入，每行包含两个字符串，以空格隔开。第一个字符串为缩略语（仅包含大写英文字符，长度不超过 10 字节），第二个字符串为日常语言（不包含空格，长度不超过 255 字节）；

3. 从第 N+2 开始到输入结束为包含缩略语的相关文档（总长度不超过 1000000 个字节）。例：

6

PS 门户搜索部

NLP 自然语言处理

PM 产品市场部

HR 人力资源部

PMD 产品推广部

MD 市场发展部

百度的部门包括 PS，PM，HR，PMD，MD 等等，其中 PS 还包括 NLP 小组。

输出要求：

输出将缩略语转换成日常语言后的文档。（将缩略语转换成日常语言，其他字符保留原样）。例：

百度的部门包括门户搜索部，产品市场部，人力资源部，产品推广部，市场发展部等等，其中门户搜索部还包括自然语言处理小组。

3. 一个猜数游戏，判断一个人反应快慢。请验证如下程序，观察 goto 的特殊作用（提示，自己编程时候尽量不要用 goto）

1. 程序分析：

2. 程序源代码：

```
#include "time.h"
#include "stdlib.h"
#include "stdio.h"
void main()
{
    char c;
    clock_t start,end;
    time_t a,b;
    double var;
    int i,guess;
    srand(time(NULL));
    printf("do you want to play it. ('y' or 'n') \n");
    loop:
    while((c=getchar())=='y')
    {
        i=rand()%100;
        printf("\nplease input number you guess:\n");
        start=clock();
        a=time(NULL);
        scanf("%d",&guess);
        while(guess!=i)
        {
            if(guess>i)
            {
                printf("please input a little
smaller.\n");
```

```

scanf("%d",&guess);
    }
    else
    {
        printf("please input a little
bigger.\n");
        scanf("%d",&guess);
    }
}
end=clock();
b=time(NULL);
printf("\1: It took you %6.3f
seconds\n",var=(double)(end-start)/1000);
printf("\1: it took you %6.3f
seconds\n\n",difftime(b,a));
if(var<15)
    printf("\1\1 You are very clever! \1\1\n\n");
else if(var<25)
    printf("\1\1 you are normal! \1\1\n\n");
else
    printf("\1\1 you are stupid! \1\1\n\n");
printf("\1\1 Congradulations \1\1\n\n");
printf("The number you guess is %d",i);
}

printf("\ndo you want to try it
again?(\\"yy\\".or.\\n\\")\n");
if((c=getch())=='y')
    goto loop;
}

```

4. 题目：利用递归函数调用方式，将所输入的 5 个字符，以相反顺序打印出来。

1. 程序分析：

2. 程序源代码：

5.

习题描述：

分子为1的分数称为埃及分数，现输入一个真分数，编写程序将该分数分解为埃及分数。如：  
 $8/11=1/2+1/5+1/55+1/110$ 。

输入描述：

输入分子和分母。

输出描述：

输出格式如下结果显示。

输入示例：

8 11

输出示例：

$8/11=1/2+1/5+1/55+1/110$

## 6. 数值截取函数

习题描述：

输入一个字符串，内有数字和非数字字符。例如：a123x456 17960 302tab5876。将其中连续的数字作为一个整数，依次存放到一维数组a中，例如123放在a[0]，456放在a[1].....统计共有多少个整数，并输出这些数。

输出描述：

首先输出个数，输出格式：num=...，第2行输出多个整数。

输入示例：

a123x456 7689?89njmk32lnk123

输出示例：

num=6

123 456 7689 89 32 123

7. 已定义一个含有 20 个元素的数组 a[]，函数 f1() 的功能是按顺序分别赋予各元素从 2 开始的偶数；函数 f2() 则按顺序每五个元素求一个平均值，并将该值存放在数组 s 中。主函数中分别调用 f1 和 f2 函数，并输出数组 a 和数组 s，数组 a 每五个元素一行；数组 s 用一行输出。

## 8. 自守数

请编写一个求自守数的函数，main 函数调用该函数求出 200000 以内的自守数

自守数是指一个数的平方的尾数等于该数自身的自然数。例如：

$$25^2=625$$

$$76^2=5776$$

$$9376^2=87909376$$

\*题目分析与算法设计

若采用“求出一个数的平方后再截取最后相应位数”的方法显然是不可取的，因为计算机无法表示过大的整数。

分析手工方式下整数平方(乘法)的计算过程，以 376 为例：

|        |                     |
|--------|---------------------|
| 376    | 被乘数                 |
| X 376  | 乘数                  |
| -----  |                     |
| 2256   | 第一个部分积=被乘数*乘数的倒数第一位 |
| 2632   | 第二个部分积=被乘数*乘数的倒数第二位 |
| 1128   | 第三个部分积=被乘数*乘数的倒数第三位 |
| -----  |                     |
| 141376 | 积                   |

本问题所关心的是积的最后三位。分析产生积的后三位的过程，可以看出，在每一次的部分积中，并不是它的每一位都会对积的后三位产生影响。总结规律可以得到：在三位数乘法中，对积的后三位产生影响的部分积分别为：

第一个部分积中：被乘数最后三位\*乘数的倒数第一位

第二个部分积中：被乘数最后二位\*乘数的倒数第二位

第三个部分积中：被乘数最后一位\*乘数的倒数第三位

将以上的部分积的后三位求和后截取后三位就是三位数乘积的后三位。这样的规律可以推广到同样问题的不同位数乘积。

按照手工计算的过程可以设计算法编写程序。

#### 9. 单科成绩分析统计程序课本

请编写一个单科成绩分析统计程序，用 `input()` 函数输入分数，`count()` 函数求各分数段人数和及格率，`avmaxmin()` 函数实现求 N 个同学的最高分，最低分，平均分，在 `main()` 函数中调用各个子函数。