**实验十　字符串**

|  |
| --- |
| 实验目标 |
| 1. 掌握字符串概念 2. 了解字符串常见错误 3. 熟悉字符串常用函数 4. 巩固指针，掌握字符串和指针间的关系 |

|  |
| --- |
| E-easy必做 N-normal必做 H-hard选做 |

1. **预习检测（调试，基本指令） E**
   1. 请下载程序stringerror.c
      1. 确保每个字符串以‘\0’结尾。
      2. 观察无’\0’结尾时，程序运行错误现象是什么
   2. 字符串输入输出常见问题

Test data测试输入：abcd a bc

Abcdefgh

观察程序的输入输出状况。替换程序中的scanf/printf为 gets/puts,用相同的测试数据，观察输入输出状况有何不同。

1. **字符串函数应用 M**

请从键盘读入3个字符串，（1）输出每个字符串的长度；（2）将字符串合并至第四个字符串 （可动态分配或分配一个很大的静态数组）后输出。

1. **高精度计算**

有些计算要求高精度，希望计算机可以计算长度超过LONG LONG类型的数据，比如几十位或者几百位。请利用字符串来试一试完成一个高精度的整数加法运算。

例如：加数“1234567891234567”被加数“2233445566778899”

分析：

1. 注意加法需要个位对齐，而数组的下标以字符串读入时，下标0对应最高位，因此最后将字符串转置即个位在下标0处
2. 字符串读入后，所有数字均为ASCII码，需要转换为对应的数字。
3. 以上两个问题解决过程中，可以将原始加数被加数字符串，转换如对应的整数数组进行存放和后续计算。
4. 思考：如何完成比如乘法之类的计算？参考highprecisioncalculate.cpp
5. 思考 H：如何完成实数的高精度计算（注意可以找到小数点后分别处理小数和整数部分）
6. **头疼的名单M**

贝小七结婚，忙坏了Beckham夫妇，现在已知来访的宾客名单 在namelist.c中，请你写程序整理名单按照字母顺序排列。

* 1. 问题分析：首先我们看到程序中宾客的已经用2维字符数组存放，我们要做的实际上就是排序而已。那么我们好好分析一下排序算法，怎么修改才可以完成任务。考虑到冒泡排序的拷贝次数更多，因此在每个数据都是字符串的情况下，效率更低，因此较好的选择是采用插入法排序。但此处我们用bubblesort为大家分析。

void BubbleSort(int srcarr[],int sortedarr[], int ilen)//传入2个2维字符数组

{

int i,j,t;//变量类型需适当变化

for(j=0; j<ilen-1; j++)//此处长度应该和2维数组的行数相关

{

for(i=0; i < ilen-1-j; i++)//此处长度应该和2维的行数相关

{

if (sortedarr[i]<sortedarr[i+1])//字符串比较用strcmp，比较对象为 //第i列一维字符串数组

{

t=sortedarr[i]; //此3处是字符串的拷贝

sortedarr[i]=sortedarr[i+1];

sortedarr[i+1]=t;

}

}

}

}

1. **迷你搜索软件 H**

大家的日常生活都离不开搜索引擎，搜索引擎结合了计算机和数学里面的好多技术，下面我们做个非常简单的模拟。假设有str1如下，要求用户键入要搜索的字符串，判断该字符串是否是str1的子字符串，如果是显示第一次匹配的下标，否则显示不是子字符串。

str1=“hello world”

Input:ell

Output: ell are found in str1 at index 1;

Input ole

Output: ole are not found in str1.

* 1. 问题分析：注意题目里用户键入的字符串长度是未知的，为了简便，你可以预定义一个稍大的静态字符数组。

在具体的算法实现过程中，请考虑以下不同情况,假设target代表要搜索的子字符串，src代表源字符串：我们知道，至少需要2重循环，外循环用于在src上逐一寻找第一个匹配字符，内循环用于在找到第一匹配字符的情况下，搜索余下字符是否匹配。

第一阶段：target和src需要比较的最大次数如何确定？即外循环次数

1. target比src长：外部需要0次循环
2. target 和src等长: 外部需要1次循环
3. target 比 src短：外部需要length(src)-length（target）次循环

这说明当比较时，如果src余下的部分小于target长度，终止搜索

第二阶段：target和src在某次具体比较中的不同情况

1. target[j] （j=0）和src[i]不相等,显然仅需要i++后继续比较，j不变
2. target[j]（j=0）和src[i]相等，显然i++，j++继续比较，由于i，j增长同步，可以写作target[j],src[i+j]
3. 在（2）基础上，target[j](j>0)和src[i+j]不相等,此时需要重新返回到src[i+1]搜索，即j回到0，同时i++，target[j] （j=0）和src[i]比较

此题的困难之处就在于如何将第二阶段用算法描述出来。第二阶段中的第1,3两个情况都需要回到外重循环，并由外循环的条件来判断外循环是否终止。

* 1. 请你自己细化算法步骤，并编程实现。

1. **截获非法申请人 M**

大家都有邮箱，[一个正常的邮箱号看起来像是这样的:sw\_xiaoming@sw.scu.edu.cn](mailto:一个正常的邮箱号看起来像是这样的:sw_xiaoming@sw.scu.edu.cn)，目前某个网站的网管发现有个非法软件在不断尝试在网站上用邮箱申请注册名，这个软件产生的部分邮箱地址不符合邮箱地址的特征，你能帮助网站的网管写个程序来做一下初步的筛查吗？合法邮箱满足以下4条规则

Rule1:只有 1个@，至少一个 .(dot)

Rule2:不能出现多个 .相连

Rule2:不能出现.@或者@.

Rule3:.和@都不能作为第一个和最后一个字符

* 1. 问题分析：规则1,和规则2不能简单的通过枚举或查找实现，请你一定先思考好如何将.和@在字符串中出现的位置描述来（即数据结构）问题，再思考如何用算法解决。
  2. 能否再想想写个自动生成邮箱的函数，来帮助你实现测试呢？
  3. 能否再想想如何过滤垃圾邮件呢？

1. **溢出攻击—来自黑客的问候 H**

在本章我们学习了很多关于字符串的函数和指令，但它们安全吗？比如scanf读入字符串大于我们程序中预定义的字符数组长度会怎么样呢？下面请看关于溢出攻击的介绍，来自百度百科<http://baike.baidu.com/link?url=_xUiB3lvKHDtpirrs4Dpv1_tEqmII2ccNhmdUw2YYgznWsHYzxnDxsh6V-HFdbCpL2rIQFf8XuQxhxqMxe6DzK>

看完后，想一想你写的程序稳定吗？

再想一想如何改进比如scanf的稳定性呢？请自己写一个函数myscanf\_s，确保用户输入的字符串不会产生溢出。你可以在网上搜索scanf\_s并参考。

http://img.proxz.xmtbang.com/?url=http%3A%2F%2Fmmbiz.qpic.cn%2Fmmbiz%2FZRL5Agu9GP8Uhict2iaunXh03icD8yUFGgZGhm8mtUkqtrKjkgCFk6iaPLxzv8cvrDre87qqIc6KfGhKVgbYCbiazJg%2F640%3Fwx_fmt%3Djpeghttp://img.proxz.xmtbang.com/?url=http%3A%2F%2Fmmbiz.qpic.cn%2Fmmbiz%2FZRL5Agu9GP8Uhict2iaunXh03icD8yUFGgZGhm8mtUkqtrKjkgCFk6iaPLxzv8cvrDre87qqIc6KfGhKVgbYCbiazJg%2F640%3Fwx_fmt%3Djpeghttp://img.proxz.xmtbang.com/?url=http%3A%2F%2Fmmbiz.qpic.cn%2Fmmbiz%2FZRL5Agu9GP8Uhict2iaunXh03icD8yUFGgZGhm8mtUkqtrKjkgCFk6iaPLxzv8cvrDre87qqIc6KfGhKVgbYCbiazJg%2F640%3Fwx_fmt%3Djpeghttp://img.proxz.xmtbang.com/?url=http%3A%2F%2Fmmbiz.qpic.cn%2Fmmbiz%2FZRL5Agu9GP8Uhict2iaunXh03icD8yUFGgZGhm8mtUkqtrKjkgCFk6iaPLxzv8cvrDre87qqIc6KfGhKVgbYCbiazJg%2F640%3Fwx_fmt%3Djpeg