

Lab 5 (上)

Lab 5 对应 lecture notes 的 Lecture 5 (数组和字符串)，训练目标是熟练使用 C++ 数组和字符串，并利用这些知识编写更多 C++ 程序。

第一部分：一维数组

Problem 1.

定义一个大小为 10 的 int 型数组 A，数组的前 3 个数是 1,-2,3，其余 7 个数字从键盘输入。编写程序完成以下内容：

1. 计算并输出数组 A 中最大数、最小数、所有数的平均值 (循环只能用一次)；
2. 将数组中的数逆序重新存放，例如，原来的顺序为 1,-2,3,8,6。要求改为 6,8,3,-2,1 (不能使用额外的数组)；
3. 定义另外一个同样大小的数组 B，将 A 中的数复制到数组 B 中；
4. 前面学到，数组访问不能越界，因此，数组 A 只能访问 A[0], A[1], ..., A[9]。请验证如果进行了非法访问，例如 A[-1], A[10], A[11]，会发生什么？

Problem 2.

在线性代数、计算几何中，向量点积是一种十分重要的运算。给定两个 n 维向量 $a=(a_1,a_2,\dots,a_n)$ 和 $b=(b_1,b_2,\dots,b_n)$ ，编写程序求点积 $a \cdot b = a_1b_1 + a_2b_2 + \dots + a_nb_n$ 。

输入

第一行是一个整数 n。 $1 \leq n \leq 1000$

第二行包含 n 个整数 a_1,a_2,\dots,a_n

第三行包含 n 个整数 b_1,b_2,\dots,b_n

相邻整数之间用单个空格隔开。每个整数的绝对值都不超过 1000

输出

一个整数，即两个向量的点积结果

Problem 3.

排序是计算机领域最重要的问题之一，有非常广泛的应用。编写程序，利用数组完成两种排序算法：

1. 从键盘输入任意个整数 (整数个数少于 50，输入时用空格隔开)，采用冒泡排序法对数字排序并按从小到大顺序输出；
2. 从键盘输入任意个整数 (整数个数少于 50，输入时用空格隔开)，采用插入排序法对数字排序并按从大到小顺序输出；
3. 对于同一组输入，统计上述两种算法执行过程中各自经过了多少次数字比较，并思考当输入的数字呈现什么特征的时候冒泡排序/插入排序需要比较的次数比较少？

提示：冒泡排序法和插入排序法请自己查阅资料，并熟练掌握至少一种排序算法的实现。

Problem 4.

查找是非常常用的操作，查找的任务是：给定一个数组和一个数，判断该数是否在数组中。编写程序分别实现以下两种场景下的查找操作：

1. 假设数组中的数是无序的;
2. 假设数组中的数是从大到小排好序的 (要求: 使用二分查找);

输入

第一行 1 个整数 (要查找的数)

第二行任意个整数 (作为数组中的数), 个数少于 50

相邻整数之间用单个空格隔开。每个整数的绝对值都不超过 1000。

输出

如果该数在数组中, 输出该数所在的数组下标 (如果有多个, 全部输出), 否则输出 No

Problem 5.

前面学过, C++的整数类型都有表示范围, 不能表示非常大的数 (例如, 超过 100 位的整数)。如果要进行大数运算, 需要特殊处理, 此类问题称为高精度运算。请利用数组、循环等知识, 实现以下高精度运算。

1. 求两个不超过 200 位的非负整数的和。

输入

有两行, 每行是一个不超过 200 位的非负整数, 没有多余的前导零

输出

一行, 即大整数的和

2. 求两个不超过 200 位的大整数的差。

输入

有两行, 每行是一个不超过 200 位的整数, 没有多余的前导零

输出

一行, 即大整数的差

3. 任意给定一个正整数 $N(N \leq 100)$, 计算 2 的 N 次方的值。

输入

正整数 N

输出

2 的 N 次方

4. 求 10000 以内 N 的阶乘。

输入

正整数 N

输出

N 的阶乘

第二部分: 字符串

注意: 完成本小题目时不允许使用 C++提供的 string 类型, 要求必须使用字符数组。

Problem 6.

编写程序, 完成以下简单字符串操作:

1. 从键盘读入一个字符串 (不含空格), 输出该字符串;
2. 从键盘读入一个字符串 (含空格), 输出该字符串;

3. 从键盘读入一个字符串，统计该字符串含数字字符的个数；
4. 从键盘读入一个字符串，将所有的小写字母变成大写字母；

Problem 7.

C++提供了一些用于字符串处理的库函数，请根据以下提示，自己学习这些库函数的使用方法，并根据学过的字符串知识自己实现这些库函数功能（[使用这些函数时请加#include<string.h>](#)）。

1. strcpy 函数

原型：strcpy(str1,str2);

功能：将字符串 str2 复制到字符串 str1 中，并覆盖 str1 原始字符串

[注意](#)：1) 字符串 str2 会覆盖 str1 中的全部字符，2) 字符串 str2 的长度不能超过 str1

例子：

```
char str1[] = "We are csdn!";
```

```
char str2[] = "Hello!";
```

```
strcpy(str1, str2); //str1 会变为 Hello!
```

自己编写程序，从键盘读入两个字符串 str1 和 str2，实现 strcpy 的功能。

输入：

第一行，字符串 str1

第二行，字符串 str2

输出：

复制之后的 str1

2. strcat 函数

原型：strcat(str1,str2);

功能：将字符串 str2 添加到字符串 str1 的尾部，也就是拼接两个字符串

[注意](#)：拼接之后的长度不能超过字符串 str1 所在数组的长度

例子：

```
char str1[20] = "We are csdn!";
```

```
char str2[] = "Hello!";
```

```
strcat(str1, str2); //str1 会变为 We are csdn!Hello!
```

自己编写程序，从键盘读入两个字符串 str1 和 str2，实现 strcpy 的功能。

输入：

第一行，字符串 str1

第二行，字符串 str2

输出：

拼接之后的 str1

3. strlen 函数

原型：strlen(str1);

功能：计算字符串 str1 的长度

[注意](#)：字符串的长度不包括'\0'

例子：

```
char str1[20] = "We are csdn!";  
int size = strlen(str1); //size 为 12
```

自己编写程序，从键盘读入 1 个字符串 str1，实现 strlen 的功能。

输入：

字符串 str1

输出：

str1 的长度

4. strcmp 函数

原型：strcmp(str1, str2);

功能：比较两个字符串，如果两个字符串相等，则返回 0；若 str1 大于 str2（指从两个字符串的第一个字符开始比较，若两个字符相同，则继续比较，若发现两个字符不相等，且 str1 中该字符的 ASCII 码大于 str2 中的，则表示 str1 大于 str2），返回一个正数；若 str1 小于 str2，返回一个负数；若字符串 str1 的长度大于 str2，且 str2 的字符与 str1 前面的字符相同，则 str1 大于 str2

例子：

```
char str1[20] = "Wearecsdn!";  
char str2[] = "Wearea!";  
int cmp = strcmp(str1, str2); //cmp 为正数
```

自己编写程序，从键盘读入两个字符串 str1 和 str2，实现 strcmp 的功能。

输入：

第一行，字符串 str1

第二行，字符串 str2

输出：

如果 str1 等于 str2，输出 0；如果 str1 大于 str2，输出 1；如果 str1 小于 str2，输出 -1；

5. atoi 函数

原型：atoi(str1);

功能：将字符串转换为对应的整数

例子：

```
char str1[20] = "123";  
int size = atoi(str1); //size 为 123
```

自己编写程序，从键盘读入 1 个字符串 str1，实现 atoi 的功能。

输入：

字符串 str1

输出：

str1 对应的整数，如果输入非法(不是一个整数)，输出 invalid；

6. atof 函数

原型：atof(str1);

功能：将字符串转换为 double 浮点数

例子：

```
char str1[20] = "123.45";
```

```
double size = atof(str1); //size 为 123.45
```

自己编写程序，从键盘读入 1 个字符串 str1，实现 atof 的功能。

输入：

字符串 str1

输出：

str1 对应的浮点数；如果输入非法(不是一个浮点数)，输出 invalid;

Problem 8.

编写程序，实现以下字符串操作。

1. 从键盘读入两个字符串 str1 和 str2，判断 str1 是否包含 str2，如果是，输出 str1 中第一次出现 str2 的起始位置（数组下标），如果不是，输出 No。

输入

第一行，字符串 str1

第二行，字符串 str2

输出

如果 str1 包含 str2，输出第一次出现 str2 的起始位置（数组下标），如果不是，输出 No

2. 给定一个单词，如果该单词以 er、ly 或者 ing 后缀结尾，则删除该后缀（题目保证删除后缀后的单词长度不为 0），否则不进行任何操作。

输入

输入一行，包含一个单词（单词中间没有空格，每个单词最大长度为 32）。

输出

输出按照题目要求处理后的单词。

3. 输入一个字符串，其中有许多连续的空格，写程序过滤掉多余的空格，只留下一个空格。
例如，"Hello world. This is C++"过滤之后变为"Hello world. This is C++"

输入

输入一行，一个字符串（长度不超过 100），句子的头和尾都没有空格。

输出

过滤之后的字符串

4. 输入一个字符串，以回车结束（字符串长度 ≤ 100 ）。该字符串由若干个单词组成，单词之间用一个或者多个空格隔开，写程序统计单词的个数。

输入

输入一行，包含多个单词的字符串

输出

字符串包含单词的个数

5. 输入一个字符串，以回车结束（字符串长度 ≤ 100 ）。该字符串由若干个单词组成，单词之间用一个或者多个空格隔开，写程序计算各个单词的长度。

输入

输入一行，包含多个单词的字符串

输出

依次输出每个单词的长度

6. 输入一个句子，将句子中的每一个单词翻转后输出。例如，输入 Hello World，输出 olleH dlroW。

输入

输入一行，为一个字符串，不超过 100 个字符，单词之间以空格隔开

输出

翻转每一个单词后的字符串，单词之间的空格需与原文一致。

7. 输入一个字符串，输出该字符串是否回文。回文是指顺序读和倒序读都一样的字符串。

输入

输入一行，为一个字符串，不超过 100 个字符

输出

如果字符串是回文，输出 yes，否则输出 no