

温州大学计算机与人工智能学院

程序设计基础(2023-2024-1) 课程作业

实验名称	字符串数组排序问题				
班 级	23大数据1	姓 名	徐王晶	学 号	23211870102
实验地点	南5B105	实验时间	2023-10-24,18:30:01	指导老师	虞铭财

一、问题编号：

0300

地址: <http://10.132.254.54/problem/300/>

二、问题描述：

给定一组字符串，按指定的排序方式输出这些字符串。
排序可是自然升序（inc）、自然逆序（dec）、
忽略大小写升序（ncinc）、忽略大小写逆序（ncdec）等。

三、输入说明：

输入有多行，第一行为一个表明排序方式的字符串，第二行为字符串的数目。

其余各行每行一个字符串（字符串的长度不超过100）。
字符串中间可能有空格，前后也可能有空格，但排序时忽略前后的空格，而输出时仍然保留。

四、输出说明：

输出也有多行，按指定的排序方式输出 那些输入的字符串。

五、输入样列：

```
ncdec
3
Hello World!
You're right!
haha! you're wrong!
```

六、输出样列：

```
You're right!
Hello World!
haha! you're wrong!
```

七、解答内容：

所用语言：

源代码：

```
001. #include <stdio.h>
002. #include <string.h>
003.
004. int str_compare(char str1[], char str2[]);
005. void upper(char str[], char new_str[]);
006.
007. int main(void)
008. {
009.     char inc[] = "inc";
010.     char dec[] = "dec";
011.     char ncinc[] = "ncinc";
012.     char ncdec[] = "ncdec";
013.     char string[100][100];
014.     char upper_string[100][100];
015.     char str[100];
016.     int n;
017.
018.
019.     scanf("%s", str);
020.
021.     scanf("%d", &n);
022.     getchar();
023.     for (int i = 0; i < n; i++)
024.     {
```

```

024.     {
025.         gets(string[i]);
026.         upper(string[i], upper_string[i]);
027.     }
028.
029.     if (strcmp(str, inc) == 0)
030.     {
031.         for (int i = 0; i < n; i++)
032.         {
033.             for (int j = 0; j < n - 1 - i; j++)
034.             {
035.                 if (str_compare(string[j], string[j+1]) > 0)
036.                 {
037.                     char temp[100];
038.                     strcpy(temp, string[j]);
039.                     strcpy(string[j], string[j + 1]);
040.                     strcpy(string[j + 1], temp);
041.                 }
042.             }
043.         }
044.     }
045.     else if (strcmp(str, dec) == 0)
046.     {
047.         for (int i = 0; i < n; i++)
048.         {
049.             for (int j = 0; j < n - 1 - i; j++)
050.             {
051.                 if (str_compare(string[j], string[j+1]) < 0)
052.                 {
053.                     char temp[100];
054.                     strcpy(temp, string[j]);
055.                     strcpy(string[j], string[j + 1]);
056.                     strcpy(string[j + 1], temp);
057.                 }
058.             }
059.         }
060.     }
061.     else if (strcmp(str, ncinc) == 0)
062.     {
063.         for (int i = 0; i < n; i++)
064.         {
065.             for (int j = 0; j < n - 1 - i; j++)
066.             {
067.                 if (str_compare(upper_string[j], upper_string[j+1]) > 0)
068.                 {
069.                     char temp[100];
070.                     strcpy(temp, string[j]);
071.                     strcpy(string[j], string[j + 1]);
072.                     strcpy(string[j + 1], temp);
073.
074.                     strcpy(temp, upper_string[j]);
075.                     strcpy(upper_string[j], upper_string[j + 1]);
076.                     strcpy(upper_string[j + 1], temp);
077.                 }
078.             }
079.         }
080.     }
081.     else if (strcmp(str, ncdec) == 0)
082.     {
083.         for (int i = 0; i < n; i++)
084.         {
085.             for (int j = 0; j < n - 1 - i; j++)
086.             {
087.                 if (str_compare(upper_string[j], upper_string[j+1]) < 0)
088.                 {
089.                     char temp[100];
090.                     strcpy(temp, string[j]);
091.                     strcpy(string[j], string[j + 1]);
092.                     strcpy(string[j + 1], temp);
093.
094.                     strcpy(temp, upper_string[j]);
095.                     strcpy(upper_string[j], upper_string[j + 1]);
096.                     strcpy(upper_string[j + 1], temp);
097.                 }
098.             }
099.         }
100.     }
101.
102.     for (int i = 0; i < n; i++)
103.     {
104.         puts(string[i]);
105.     }
106.
107.     return 0;
108. }
109.
110. int str_compare(char str1[], char str2[])
111. {
112.     int i = 0;
113.     int j = 0;
114.     int flag = 0;
115.     while (str1[i] == ' ')

```

```

116.     {
117.         i++;
118.     }
119.     while (str2[j] == ' ')
120.     {
121.         j++;
122.     }
123.     while (str1[i] != '\0' && str2[j] != '\0')
124.     {
125.         if (str1[i] > str2[j])
126.         {
127.             return 1;
128.         }
129.         else if (str1[i] < str2[j])
130.         {
131.             return -1;
132.         }
133.         i++;
134.         j++;
135.     }
136. }
137.
138. void upper(char str[], char new_str[])
139. {
140.     int i = 0;
141.     while (str[i] != '\0')
142.     {
143.         if (str[i] >= 'a' && str[i] <= 'z')
144.         {
145.             new_str[i] = str[i] - 32;
146.         }
147.         else
148.         {
149.             new_str[i] = str[i];
150.         }
151.         i++;
152.     }
153.     new_str[i] = '\0';
154. }

```

八、判题结果

AC - 答案正确

判题结果补充说明：

test id: 568,result:AC, usedtime:0MS, usedmem:1036KB,score:25 test id: 708,result:AC, usedtime:0MS, usedmem:1036KB,score:25 test id: 709,result:AC, usedtime:0MS, usedmem:1036KB,score:25 test id: 710,result:AC, usedtime:0MS, usedmem:1036KB,score:25