

更多企业校园招聘笔试面试题合集下载: <http://bimian.xuanjianghui.com.cn/>

一、题目描述 (60 分):

通过键盘输入一串小写字母(a~z)组成的字符串。请编写一个字符串过滤程序，若字符串中出现多个相同的字符，将非首次出现的字符过滤掉。

比如字符串“abacacde”过滤结果为“abcde”。

要求实现函数: void stringFilter(const char *pInputStr, long lInputLen, char *pOutputStr);

【输入】 pInputStr: 输入字符串

lInputLen: 输入字符串长度

【输出】 pOutputStr: 输出字符串，空间已经开辟好，与输入字符串等长;

【注意】只需要完成该函数功能算法，中间不需要有任何 IO 的输入输出

示例

输入: “deefd” 输出: “def”

输入: “afafafaf” 输出: “af”

输入: “pppppppp” 输出: “p”

main 函数已经隐藏, 这里保留给用户的测试入口, 在这里测试你的实现函数, 可以调用 printf 打印输出

当前你可以使用其他方法测试, 只要保证最终程序能正确执行即可, 该函数实现可以任意修改, 但是不要改变函数原型。一定要保证编译运行不受影响。

二、题目描述 (40 分):

通过键盘输入一串小写字母(a~z)组成的字符串。请编写一个字符串压缩程序, 将字符串中连续出席的重复字母进行压缩, 并输出压缩后的字符串。

压缩规则:

1、仅压缩连续重复出现的字符。比如字符串“abcabc”由于无连续重复字符, 压缩后的字符串还是“abcabc”。

2、压缩字段的格式为“字符重复的次数+字符”。例如: 字符串“xxxxyyyyyyyz”压缩后就成为“3x6yz”。

要求实现函数:

void stringZip(const char *pInputStr, long lInputLen, char *pOutputStr);

【输入】 pInputStr: 输入字符串

lInputLen: 输入字符串长度

【输出】 pOutputStr: 输出字符串, 空间已经开辟好, 与输入字符串等长;

【注意】只需要完成该函数功能算法, 中间不需要有任何 IO 的输入输出

示例

输入: “cccddecc” 输出: “3c2de2c”

输入: “adeff” 输出: “adeff”

输入: “ppppppppp” 输出: “8p”

三、题目描述 (50 分):

通过键盘输入 100 以内正整数的加、减运算式, 请编写一个程序输出运算结果字符串。

输入字符串的格式为: “操作数 1 运算符 操作数 2”, “操作数”与“运算符”之间以一个空格隔开。

补充说明:

1、操作数为正整数, 不需要考虑计算结果溢出的情况。

2、若输入算式格式错误, 输出结果为“0”。

要求实现函数:

更多企业校园招聘笔试面试题合集下载: <http://bimian.xuanjianghui.com.cn/>

```
void arithmetic(const char *pInputStr, long lInputLen, char *pOutputStr);
```

【输入】 pInputStr: 输入字符串

lInputLen: 输入字符串长度

【输出】 pOutputStr: 输出字符串, 空间已经开辟好, 与输入字符串等长;

【注意】只需要完成该函数功能算法, 中间不需要有任何 IO 的输入输出

示例

输入: "4 + 7" 输出: "11"

输入: "4 - 7" 输出: "-3"

输入: "9 ++ 7" 输出: "0" 注: 格式错误

```
[cpp] view plaincopyprint?
```

```
//////////////////////////////////////华为第一题 19:19-19:36 17分钟
```

```
#include <iostream>
```

```
#include <cassert>
```

```
using namespace std;
```

```
bool g_flag[26];
```

```
void stringFilter(const char *pInputStr, long lInputLen, char *pOutputStr)
```

```
{
```

```
    assert(pInputStr != NULL);
```

```
    int i = 0;
```

```
    if (pInputStr == NULL || lInputLen <= 1)
```

```
    {
```

```
        return;
```

```
    }
```

```
    const char *p = pInputStr;
```

```
    while(*p != '\0')
```

```
    {
```

```
        if (g_flag[(*p - 'a')])
```

```
        {
```

```
            p++;
```

```
        }else{
```

```
            pOutputStr[i++] = *p;
```

```
            g_flag[*p - 'a'] = 1;
```

```
            p++;
```

```
        }
```

```
    }
```

```
    pOutputStr[i] = '\0';
```

```
}
```

```
int main()
```

```
{
```

```
    memset(g_flag,0,sizeof(g_flag));
```

```
    char input[] = "abacacde";
```

更多企业校园招聘笔试面试题合集下载: <http://bimian.xuanjianghui.com.cn/>

```
char *output = new char[strlen(input) + 1];
stringFilter(input,strlen(input),output);
cout<<output<<endl;
delete output;
return 0;
}
//////////////////////////////////////华为第一题 c 实现
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <assert.h>

//bool g_flag[26];
void stringFilter(const char *pInputStr, long lInputLen, char *pOutputStr)
{
    char flag[26];
    memset(flag, 0, 26);
    assert(pInputStr != NULL && lInputLen != 0);
    int i = 0;

    const char *p = pInputStr;
    while(*p != '\0')
    {
        if (flag[(*p - 'a')])
        {
            p++;
        }else{
            pOutputStr[i++] = *p;
            flag[*p - 'a'] = 1;
            p++;
        }
    }
    pOutputStr[i] = '\0';
}
int main()
{
    char input[] = "a";
    char output[sizeof(input)];
    printf("%d\n", strlen(input));
    stringFilter(input,strlen(input),output);
    printf("result:%s\n", output);
    return 0;
}
[cpp] view plaincopyprint?
```

更多企业校园招聘笔试面试题合集下载: <http://bimian.xuanjianghui.com.cn/>

////////////////////////////////////华为第二题 19:40 - 20:10 中间耽误 3 分钟

```
#include <iostream>
#include <cassert>

using namespace std;

void stringZip(const char *pInputStr, long lInputLen, char *pOutputStr)
{
    const char *p = pInputStr;
    int num = 1;
    int i = 0;
    p++;
    while(*p != NULL)
    {
        while(*p == *(p-1)&& *p != NULL)
        {
            num++;
            p++;
        }
        if (num > 1)
        {
            int size = 0;
            int temp = num;
            while(num)           //计算位数
            {
                size++;
                num /= 10;
            }
            num = 1;

            for (int j = size; j > 0; j--)
            {
                pOutputStr[i+j-1] = '0'+ temp%10;
                temp /= 10;
            }
            i +=size;
            pOutputStr[i++] = *(p-1);
            p++;
        }else{
            pOutputStr[i++] = *(p-1);
            p++;
        }
    }
    pOutputStr[i] = '\0';
}
```

更多企业校园招聘笔试面试题合集下载: <http://bimian.xuanjianghui.com.cn/>

```
}  
  
int main()  
{  
    char input[] = "cccddecc";  
    char *output = new char[strlen(input) + 1];  
    stringZip(input, strlen(input), output);  
    cout << output << endl;  
    return 0;  
}
```

//////////////////////////////////////华为第三题 20:29 - 20:40

```
#include <iostream>  
  
using namespace std;  
  
void arithmetic(const char *pInputStr, long lInputLen, char *pOutputStr)  
{  
    const char *input = pInputStr;  
    char *output = pOutputStr;  
    int sum = 0;  
    int operator1 = 0;  
    int operator2 = 0;  
    char *temp = new char[5];  
    char *ope = temp;  
    while(*input != ')') //获得操作数 1  
    {  
        sum = sum*10 + (*input++ - '0');  
    }  
    input++;  
    operator1 = sum;  
    sum = 0;  
  
    while(*input != ')')  
    {  
        *temp++ = *input++;  
    }  
  
    input++;  
    *temp = '\0';  
  
    if (strlen(ope) > 1 )  
    {
```

更多企业校园招聘笔试面试题合集下载: <http://bimian.xuanjianghui.com.cn/>

```
        *output++ = '0';
        *output = '\0';
        return;
    }

    while(*input != '\0') //获得操作数 1
    {
        sum = sum*10 + (*input++ - '0');
    }
    operator2 = sum;
    sum = 0;

    switch (*ope)
    {
    case '+': itoa(operator1+operator2,pOutputStr,10);
        break;
    case '-': itoa(operator1-operator2,pOutputStr,10);
        break;
    default:
        *output++ = '0';
        *output = '\0';
        return;
    }
}

int main()
{
    char input[] = "4 - 7";
    char output[] = "    ";
    arithmetic(input,strlen(input),output);
    cout<<output<<endl;
    return 0;
}
```