

Koishi 与字符串

题意简述

给出一个字符串，求其中最长的美丽子串的长度。子串指的是在字符串中连续的一段。美丽子串指的是这个子串中，先出现数字，后出现字母，并且字母和数字都是连续的，字母不区分大小写。多组测试数据。

题目解析

没什么好说的，就是按部就班一步一步来实现。

我们用一个计数变量 `cnt` 来记录当前的美丽字符串长度，用 `ans` 来记录答案（最长的美丽字符串长度）。具体分为以下几步：

1. 读入一个字符串；
2. 从左到右扫描字符串；（循环语句）
3. 如果遇到数字，则判断前面有没有遇到字母，如果有则计算 `ans = max(cnt, ans)`，之后清空 `cnt`。如果前面什么都没遇到则直接记入，如果前面有遇到别的数字则判断是否是连续的数字；
4. 如果遇到字母，先转化统一大小写，然后判断前面有没有遇到其他字母。如果有则判断是否为连续的字母。如果前面遇到的是数字也可以直接记入答案。
5. 如果遇到其他字符和其他不合法的情况，计算 `ans = max(cnt, ans)`，之后清空 `cnt`。
6. 在最后结束扫描的时候再计算一次 `ans = max(cnt, ans)`。

在代码实现的处理上，我用了个小技巧：把其他字符表示为 0，数字字符标识为 1，字母标识为 2。如此操作后只需要判断下一个字符的数字类型是不是不为 0，且大于等于上一个字符的类型即可。节省了一些负责的判断过程。

最后就是要注意，题目有多组测试数据。在处理下个字符串之前，`cnt, ans` 这些变量要记得清空。

代码

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define qmx(a,b) a<b?b:a

int get_type(char ch){
    if(!isdigit(ch)&&!isalpha(ch))
        return 0;        //不是数字不是字母返回0
    else{
        if(isdigit(ch))
            return 1;        //是数字返回1
        if(isalpha(ch))
            return 2;        //是字母返回2
    }
}

char str[1000006];

int main(){
    char c1,c2;
    int test;
```

```

scanf("%d", &test);
int ls,ans,cnt;
while(test--){
    ans=cnt=0;
    scanf("%s", str);
    ls=strlen(str);
    if(!get_type(str[0])) //判断第一个字符的类型
        cnt=0;
    else
        cnt=1;
    for(int t1,t2,i=1;i<ls;i++){
        t1=get_type(str[i-1]); //前一个字符的类型
        t2=get_type(str[i]);   //当前字符的类型
        if(!t2){               //当前字符不是数字不是字母，更新 ans，重置 cnt
            ans=qmx(ans,cnt);
            cnt=0;
            continue;
        }
        if(t1==t2){           //类型相同
            c1=str[i-1];
            c2=str[i];
            if(t1==2){         //都是字母，进行大小写转化（这里统一转成小写字母）
                if(isupper(c1))
                    c1=c1-'A'+'a';
                if(isupper(c2))
                    c2=c2-'A'+'a';
            }
            if(c1+1==c2)       //判断是否连续，是则 cnt + 1，否则更新 ans，重置 cnt
                cnt++;
            else{
                ans=qmx(ans,cnt);
                cnt=1;
            }
        }else{                //类型不同
            if(t2>t1){          //只能是 t2 = 2, t1 = 1的时候满足这种情况，即数字在前，
字母在后
                cnt++;
            }else{              //否则更新 ans，重置 cnt
                ans=qmx(ans,cnt);
                cnt=1;
            }
        }
    }
    ans=qmx(cnt,ans);          //最后也要记得更新 ans
    printf("%d\n",ans);
}
return 0;
}

```

补充

把字符类型映射成 0, 1, 2 这样的数字，运用了一点 **哈希** 算法的思想，感兴趣的同学可以去了解一下。

