

新哈希函数

限制

- 1000 ms
 - 32767 KB
-

小明设计了一个哈希函数，将一个长度为 k 的字符串转成一个长度为 32 的字符串。这个哈希函数 f 的设计如下：

- 声明一个长度为 32 的数组 `arr`，并将其中元素全部初始化为 0。
- 取出每一位的 ASCII 值，将长度为 k 的字符串中第 i 位的 ASCII 码加进 `arr[i % 32]` 里 ($1 \leq i \leq k$)。
- 声明一个长度为 32 的数组 `bits`，令 `bits[j]` 为 `arr[31 - j]` 与 `arr[j] << 1` 的值进行了按位异或运算后得到的结果 ($0 \leq j \leq 31$)。
- 计算出 `bits[j] % 85 + 34` 并将该十进制数在 ASCII 码中对应的字符输出到结果字符串的第 $j + 1$ 位 ($0 \leq j \leq 31$)。

请实现一个程序，当输入一个字符串 s 后，输出哈希函数的结果 $f(s)$ 。

输入格式

输入为一行，包括一个长度为 k 的字符串 ($32 < k < 500$)，这个字符串由大写字母、小写字母和数字组成（不含空格）。

输出格式

输出为一行，为一个长度为 32 的字符串哈希结果

习题目的

- 练习左移运算
 - 练习使用异或运算
 - 练习使用类型转换处理 `int` 和 `char` 类型的关系
-

初始化代码

```
#include <stdio.h>
int main() {

    return 0;
}
```

样例输入

```
123456789012345678901234567890123
```

样例输出

```
"p*+, )&'ebst*+, )&'ebst*+, )&'eb&r
```

提示信息

你的程序会被测评机输入各种不同的，但是符合描述的输入数据（所以你需要用 `scanf` 接受输入）。它之后会对你的输出和预期的标准输出进行比对，如果你在任何一个给出的输入上都给出了正确的输出，你就会通过这一测试。

同时，请注意，在测试中使用样例输入得到正确的样例输出结果也不一定代表程序就完备了。提交前，你最好再自己多设计几组符合题目要求的输入然后计算得出结果，与输出值比较，看看是不是每一次都相同。

请注意，不要额外输出任何的内容。任何所谓的“友好提示信息”，在测评机来看都是由于程序设计错误而输出的错误内容。