第一题

题目描述

输入两个字母串，将两个字母串都包含的字母用’\_'替换后，输出两个字母串的剩余部分。

输入描述:输入两个字符串，字符串最大长度为100。字符串只包含字母，不可能为空串，区分大小写。

输出描述:按字符串顺序输出处理后的字符串

示例：

输入

abcd

bdef

1

2

输出

a\_c\_

\_\_ef

1

2

解题思路

采用哈希思想。首先遍历字母串a，记录其中出现的字母，随后遍历字母串b，将重复出现的字母标记为100并替换成\_，随后再遍历a，替换重复出现的字母。

参考代码

#include<iostream>

using namespace std;

const int SIZE = 100;

int main()

{

char a[SIZE];

char b[SIZE];

int low\_alpha[26] = { 0 };

int high\_alpha[26] = { 0 };

cin.getline(a, SIZE);

cin.getline(b, SIZE);

int i = 0; int j = 0;

while (a[i] != '\0')

{

if (a[i] >= 97)

low\_alpha[a[i] - 97]++;

else

high\_alpha[a[i] - 65]++;

i++;

}

while (b[j] != '\0')

{

if (b[j] >= 97)

{

if (low\_alpha[b[j] - 97] > 0)

{

low\_alpha[b[j] - 97] = 100;

b[j] = '\_';

}

else

low\_alpha[b[j] - 97]++;

}

else

{

if (high\_alpha[b[j] - 65] > 0)

{

high\_alpha[b[j] - 65] = 100;

b[j] = '\_';

}

else

high\_alpha[b[j] - 65]++;

}

j++;

}

i = 0;

while (a[i] != '\0')

{

if (a[i] >= 97)

{

if (low\_alpha[a[i] - 97] ==100)

{

a[i] = '\_';

}

}

else

if(high\_alpha[a[i] - 65]==100)

a[i] = '\_';

i++;

}

cout << a << endl;

cout << b << endl;

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

运行结果

第二题

题目描述

一本正式出版的图书都有一个ISBN号码与之对应，ISBN码包括9位数字、1位识别码和3位分隔符，其规定格式如“x-xxx-xxxxx-x”，其中符号“-”是分隔符（键盘上的减号），最后一位是识别码，例如0-670-82162-4就是一个标准的ISBN码。ISBN码的首位数字表示书籍的出版语言，例如0代表英语；第一个分隔符“-”之后的三位数字代表出版社，例如670代表维京出版社；第二个分隔之后的五位数字代表该书在出版社的编号；最后一位为识别码。识别码的计算方法如下：

首位数字乘以1加上次位数字乘以2……以此类推，用所得的结果mod 11，所得的余数即为识别码，如果余数为10，则识别码为大写字母X。

例如ISBN号码0-670-82162-4中的识别码4是这样得到的：对067082162这9个数字，从左至右，分别乘以1，2，…，9，再求和，即

0×1+6×2+……+2×9=158 0×1+6×2+……+2×9=1580×1+6×2+……+2×9=158，

然后取158 mod 11的结果4作为识别码。

你的任务是编写程序根据输入的ISBN号码的前3段，计算出识别码，并输出完整的ISBN码。

输入描述:

为一个ASCII字符串。内容为ISBN码的前三段，以上面为例，就是0-670-82162。

输出描述:

若判断输入为合法的字符串，则计算出识别码，并输出完整的ISBN码；

若输入不合法，则输出字符串”ERROR”；

示例1

输入：

0-670-82162

1

输出：

0-670-82162-4

1

解题思路

逻辑较为简单，仅给出合理输入的实现代码

参考代码

isbn\_ini=input()

isbn=isbn\_ini.replace("-",'')

sums=0

for i in range(len(isbn)):

sums+=eval(isbn[i])\*(i+1)

if sums%11==10:

last="X"

else:

last=str(sums%11)

print(isbn\_ini+'-'+last)

1

2

3

4

5

6

7

8

9

10

第三题

题目描述

主机名由多级域名组成，自右向左，依次是顶级域名、二级域名、三级域名……以此类推

例，主机名：google.com.hk

　　hk是顶级域名

　　com是二级域名

　　google是三级域名

1

2

3

4

现在我们需要实现一个主机名的排序功能

排序规则

主机名按照域名等级排序，即先按照顶级域名排序，顶级域名相同的再按照二级域名排序，顶级和二级域名均相同的再按照三级域名排序，以此类推，直到整个主机名排序完毕

如果短主机名是由长主机名从顶级域名开始的连续一个或多个域名组成，短主机名排在长主机名前面。例：google.com 排在gmail.google.com 之前

每一级域名按照字典顺序排序，字典顺序定义见下面：

（1）两个单词(字母按照自左向右顺序)先以第一个字母作为排序的基准，如果第一个字母相同，就用第二个字母为基准，如果第二个字母相同就以第三个字母为基准。依此类推，如果到某个字母不相同，字母顺序在前的那个单词顺序在前。

例：abc 排在 abf 之前

（2）如果短单词是长单词从首字母开始连续的一部分，短单词顺序在前。例：abc 排在 abcd 之前

输入确保符合以下规则（无需检查）

主机名以字符串形式给出，非空串

主机名中仅包含小写英文字母和分隔符’.’

主机名中没有连续的’.’，不以’.’开始，也不以’.’结束

主机名不存在重复

示例1

输入

mail.huawei.com|huawei.com|teltalk.org|google.com.hk|imail.huawei.com

1

输出

huawei.com|imail.huawei.com|mail.huawei.com|google.com.hk|teltalk.org

1

解题思路

建立域名类，重写排序

参考代码

class Domain:

def \_\_init\_\_(self,s):

ls=s.split(".")

self.top=ls[-1]

self.num=len(ls)

self.mid=''

self.third=''

if self.num==2:

self.mid=ls[0]

elif self.num==3:

self.mid = ls[1]

self.third=ls[0]

else:

pass

def \_\_gt\_\_(self, other):

if self.top!=other.top:

return self.top>other.top

else:

if self.mid!=other.mid:

return self.mid>other.mid

else:

if self.num!=other.num:

return self.num<other.num

else:

return self.third>other.third

def \_\_str\_\_(self):

if self.num==3:

return self.third+'.'+self.mid+'.'+self.top

elif self.num==2:

return self.mid+'.'+self.top

else:

return self.top

name=input().split("|")

domain=[Domain(x) for x in name]

domain.sort()

ans=list(map(str,domain))

print("|".join(ans))