homework06

2018 年 4 月 11 日

0.1 输出 1~100 之间不能被 7 整除的数，每行输出 10 个数字，要求应用字符串格式化方法（任何一种均可）美化输出格式。

```
In [2]: #************homework06_1******************
        cnt = 0
        for i in range(101):
            if i % 7 != 0:
                print("{:>4}".format(i),end = "")
                cnt += 1
            if cnt ==10:
                print()
                cnt = 0
```

```
   1   2   3   4   5   6   8   9  10  11
  12  13  15  16  17  18  19  20  22  23
  24  25  26  27  29  30  31  32  33  34
  36  37  38  39  40  41  43  44  45  46
  47  48  50  51  52  53  54  55  57  58
  59  60  61  62  64  65  66  67  68  69
  71  72  73  74  75  76  78  79  80  81
  82  83  85  86  87  88  89  90  92  93
  94  95  96  97  99 100
```

0.2 假定 s 是小写字母的字符串。编写程序，输出 s 的最长子串（子串必须是字母顺序）。

- 若 s = 'azcbobobegghakl'，则应输出：Longest substring in alphabetical order is: beggh
- 若 s = 'abcbcd'，则应输出第一个子串：Longest substring in alphabetical order is: abc

```
In [12]: #************homework06_2******************
         import re
```

```
        def subString(s):
            s_find = "0"
            for i in range(len(s)-1):
                if s[i + 1] >= s[i]:
                    s_find = s_find + str(int(s_find[i])^0)
                else:
                    s_find = s_find + str(int(s_find[i])^1)

            match_0 = re.findall(r'0+',s_find)
            match_1 = re.findall(r'1+',s_find)

            max_temp = max(match_0 + match_1, key = len)

            pos = s_find.find(max_temp)
            s_sub = s[pos : pos + len(max_temp)]
            return s_sub

        if __name__ == "__main__":
            s1 = "azcbobobegghakl"
            s2 = "abcbcd"
            s1_sub = subString(s1)
            s2_sub = subString(s2)
            print("The longest substring of '{}' in alphabetical order is: '{}'"\
                  .format(s1,s1_sub))
            print("The longest substring of '{}' in alphabetical order is: '{}'"\
                  .format(s2,s2_sub))
```

```
The longest substring of 'azcbobobegghakl' in alphabetical order is: 'beggh'
The longest substring of 'abcbcd' in alphabetical order is: 'abc'
```

0.3 编写函数 *OnlyCharNum(s)*，对于给定的字符串只显示字母和数字

- 若 s= "a000 aa-b"，则程序输出：a000aab

```
In [10]: #************homework06_3*******************
         import re
```

```
        def OnlyCharNum(s):
            match = re.findall(r'\w+', s)
            return "".join(match)

        if __name__ == "__main__":
            s = "a000 aa-b"
            print(OnlyCharNum(s))
```

```
a000aab
```

0.4 假定给定两个字符串（它们可能是空串）s1 和 s2，要将这两个串合并在一起。要求是：s1 串的第一个字符与 s2 串的第一个字符连在一起，后续以此类推。若一个串的长度大于另一个串的长度，则保留长串的剩余字符，即连接到新串的末尾即可。编写函数 *laceStrings(s1,s2)*，完成上述功能。

- 若 s1='abcd'、s2='efghi'，则新串为：'aebfcgdhi'。

```
In [11]: ##########homework06_4##################
        def laceString(s1, s2):

            s_lace = ""
            len1 = len(s1)
            len2 = len(s2)

            for i in range(min(len1,len2)):
                s_lace += s1[i]
                s_lace += s2[i]
            if len1 > len2:
                s_lace += s1[len2 : ]
            elif len2 > len1:
                s_lace += s2[len1 : ]

            return s_lace


        if __name__ == "__main__":
```

```
for i in range(3):
    s1 = input("Please input the 1st string: ")
    s2 = input("Please input the 2nd string: ")
    s_lace = laceString(s1, s2)
    print("s1 = {}\ns2 = {}".format(s1,s2))
    print("The laceString is {}.".format(s_lace))
    print()
```

```
Please input the 1st string: aapple
Please input the 2nd string: bdlise
s1 = aapple
s2 = bdlise
The laceString is abadplpilsee.

Please input the 1st string: wdiwiie
Please input the 2nd string: abliejlls
s1 = wdiwiie
s2 = abliejlls
The laceString is wadbilwiieijells.

Please input the 1st string: abijekekk
Please input the 2nd string: zbiejd
s1 = abijekekk
s2 = zbiejd
The laceString is azbbiijeejkdekk.
```