###part1

1. 成员和身份运算符如何使用

成员运算符 *in* 和 *not in* (针对于容器型数据)  
*in* 判断在里面  
*not in* 判断不在里面  
判断字典的时候是判断键，不是值  
  
身份运算符 *is* | *is not* 检测两个数据在内存中是否是同一个地址  
整型 -5 ~ 正无穷，id一致  
浮点型 非负数，id一致  
复数 在实数 + 虚数这样的结构中永远不一样 但如果只有虚数 在值相同情况下一样  
bool 在布尔值相同的情况下 id一样  
str 字符串而言，字符串值相同情况下，id一致  
剩下的容器类型数据无论什么值都不一样(空元组例外)

1. 逻辑运算符优先级?逻辑短路在什么情况下发生?

逻辑优先级：() > *not* > *and* > *or*逻辑短路：  
一个or或者一个and 情况下  
*True or* something  
*False and* something

1. 优先级最高和最低的运算符是?

整体一元运算符优先级 > 二元运算符  
所有运算符当中 \*\* 幂运算优先级最高  
所有运算符当中 = 赋值运算符优先级最低

1. 左移右移后的值如何计算?按位非公式?

左移：n << m n乘以2的m次幂  
右移：n >> m n除以2的m次幂  
~ 按位非 - ( n+1 )

1. res = 17>15 or 78<11 or 7 and 8 and not True is True res=?

*True*

6.计算如下值

1).6 or 2 > 1

6

2).3 or 2 > 1

3

3).0 or 5 > 4

*True*

4).5 < 4 or 3

3

5).2 > 1 or 6

*True*

6).3 and 2 > 1

*True*

7).0 and 3 > 1

0

8).2 > 1 and 3

3

9).3 > 1 and 0

0

10).3 > 1 and 2 or 2 < 3 and 3 and 4 or 3 > 2

2

11)not 2 > 1 and 3 < 4 or 4 > 5 and 2 > 1 and 9 > 8 or 7 < 6

*False*

7.利用if语句写出猜大小的游戏：

设定一个理想数字比如：66，让用户输入数字，如果比66大，则显示猜测的结果大了；如果比66小，则显示猜测的结果小了;只有等于66，显示猜测结果正确。

res = input('请输入一个数字：').strip()  
*if* int(res) == 66:  
 print('猜对了') *elif* int(res) > 66:  
 print('太大了')  
*else*:  
 print('太小了')

8.如果小于10, 提示小屁孩, 如果大于10, 小于20,提示青春期叛逆的小屁孩.

如果大于20, 小于30. 提示开始定性, 开始混社会的小屁孩儿,

如果大于30, 小于40. 提示看老大不小了, 赶紧结婚小屁孩儿.

如果大于40, 小于50. 提示家里有个不听话的老屁孩儿.

如果大于50, 小于60. 提示自己马上变成不听 话的老屁孩儿.

如果大于60, 小于70. 提示活着还不错的老屁孩儿.

如果大于70, 小于 90. 提示人生就快结束了的一个老屁孩儿.

如果大于90以上. 提示. 再见了这个世界.

res = int(input('输入你的年龄').strip())  
*if* res < 10:  
 print('小屁孩')  
*elif* 10 <= res < 20:  
 print('开始混社会的小屁孩儿')  
*elif* 20 <= res < 30:  
 print('开始定性, 开始混社会的小屁孩儿')  
*elif* 30 <= res < 40:  
 print('看老大不小了, 赶紧结婚小屁孩儿')  
*elif* 40 <= res < 50:  
 print('家里有个不听话的老屁孩儿')  
*elif* 50 <= res < 60:  
 print('自己马上变成不听 话的老屁孩儿')  
*elif* 60 <= res < 70:  
 print('活着还不错的老屁孩儿')  
*elif* 70 <= res < 90:  
 print('人生就快结束了的一个老屁孩儿')  
*elif* res >= 90:  
 print('再见了这个世界')

9.提示用户输入麻花藤. 判断用户输入的对不对. 如果对, 提示真聪明, 如果不对, 提示你 是二货么

res = input('腾讯老板的名字').strip()  
*if* res == '麻花藤':  
 print('真聪明')  
*else*:  
 print('你是二货么')

10.输出 1-100 内的所有奇数

*for* i *in* range(1, 101, 2):  
 print(i)

11.输出 1-100 内的所有偶数

*for* i *in* range(2, 101, 2):  
 print(i)

12.求1-2+3-4+5 ... 99的所有数的和

count = 0  
*for* i *in* range(1, 100):  
 *if* i % 2 != 0:  
 count += i  
 *elif* i % 2 == 0:  
 count -= i  
print(count)

13.用户登陆(三次输错机会)且每次输错误时显示剩余错误次数

(提示：使用字符串格式化)

username = 'aaa'  
password = '123'  
*for* i *in* range(3):  
 usr = input('请输入用户名：').strip()  
 pwd = input('请输入密码：').strip()  
 *if* usr == username *and* pwd == password:  
 print('登录成功')  
 *break  
 else*:  
 print('密码错误，剩余%d次机会' % (2-i))

###part2

1.有变量name = "aleX leNb" 完成如下操作：

移除 name 变量对应的值两边的空格,并输出处理结果

1. 移除name变量左边的"al"并输出处理结果

print(name[2:])

1. 移除name变量右面的"Nb",并输出处理结果

print(name[:-2])

1. 移除name变量开头的a"与最后的"b",并输出处理结果

print(name[1:-1])

1. 判断 name 变量是否以 "al" 开头,并输出结果

print(name.startswith('al'))

1. 判断name变量是否以"Nb"结尾,并输出结果

print(name.endswith('Nb'))

1. 将 name 变量对应的值中的 所有的"l" 替换为 "p",并输出结果

print(name.replace('l', 'p'))

1. 将name变量对应的值中的第一个"l"替换成"p",并输出结果

print(name.replace('l', 'p', 1))

1. 将 name 变量对应的值根据 所有的"l" 分割,并输出结果。

print(name.split('l'))

1. 将name变量对应的值根据第一个"l"分割,并输出结果。

print(name.split('l', 1))

1. 将 name 变量对应的值变大写,并输出结果

print(name.upper())

1. 将 name 变量对应的值变小写,并输出结果

print(name.lower())

1. 将name变量对应的值首字母"a"大写,并输出结果

print(name.capitalize())

1. 判断name变量对应的值字母"l"出现几次，并输出结果

print(name.count('l'))

1. 如果判断name变量对应的值前四位"l"出现几次,并输出结果

print(name[:4].count('l'))

1. 从name变量对应的值中找到"N"对应的索引(如果找不到则报错)，并输出结果

print(name.index('N'))

1. 从name变量对应的值中找到"N"对应的索引(如果找不到则返回-1)输出结果

print(name.find('N'))

1. 从name变量对应的值中找到"X le"对应的索引,并输出结果

print(name.index('X le'))  
print(name.find('X le'))

1. 请输出 name 变量对应的值的第 2 个字符?

print(name[2])

1. 请输出 name 变量对应的值的前 3 个字符?

print(name[2])

1. 请输出 name 变量对应的值的后 2 个字符?

print(name[-2:])

1. 请输出 name 变量对应的值中 "e" 所在索引位置?

print(name.index('e')) # 查第一个  
  
# 枚举，查所有e的索引  
*for* index, i *in* enumerate(name):   
 *if* i == 'e':  
 print(index)

2.实现一个整数加法计算器(两个数相加)：

如：content = input("请输入内容:") 用户输入：5+9或3+ 9或5 + 6，然后进行分割再进行计算

content = input('请输入内容:').strip()  
li = content.split('+')  
print(li)  
res = int(li[0].strip()) + int(li[1].strip())  
print(res)

3.升级题：实现一个整数加法计算器（多个数相加）：

如：content = input("请输入内容:") 用户输入：5+9+6 +12+ 13，然后进行分割再进行计算。

content = input('请输入内容:').strip()  
li = content.split('+')  
print(li)  
res = 0  
*for* i *in* li:  
 res += int(i)  
print(res)

4.计算用户输入的内容中有几个整数（以个位数为单位）。

如：content = input("请输入内容：") # 如fhdal234slfh98769fjdla

content = input('请输入内容：').strip()  
count = 0  
*for* i *in* content:  
 *if* i.isdigit():  
 count += 1  
print(count)

5.等待用户输入数据,检测用户输入内容中是否包含敏感字符？

如果存在敏感字符提示"存在敏感字符请重新输入"，

并允许用户重新输入打印。

(敏感字符：小粉嫩,大铁锤)

li = ['小粉嫩', '大铁锤']  
*while* 1:  
 res = input('>>>').strip()  
 *if* res.upper() == 'Q': *break  
 for* i *in* li:  
 *if* i *in* res:  
 print('存在敏感字符，请重新输入')  
 *break  
 else*:  
 print('没有敏感字符')

6.字符串格式化：

等待用户输入名字、地点、爱好，根据用户的名字和爱好

拼装数据打印：

敬爱可亲的xxx，

最喜欢在xxx地方xxx

name = input('请输入名字：')  
site = input('请输入地点：')  
hobby = input('请输入爱好：')  
print('敬爱可亲的%s，最喜欢在%s地方干%s' % (name, site, hobby))