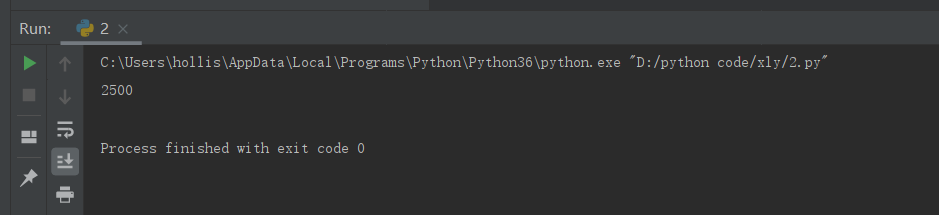
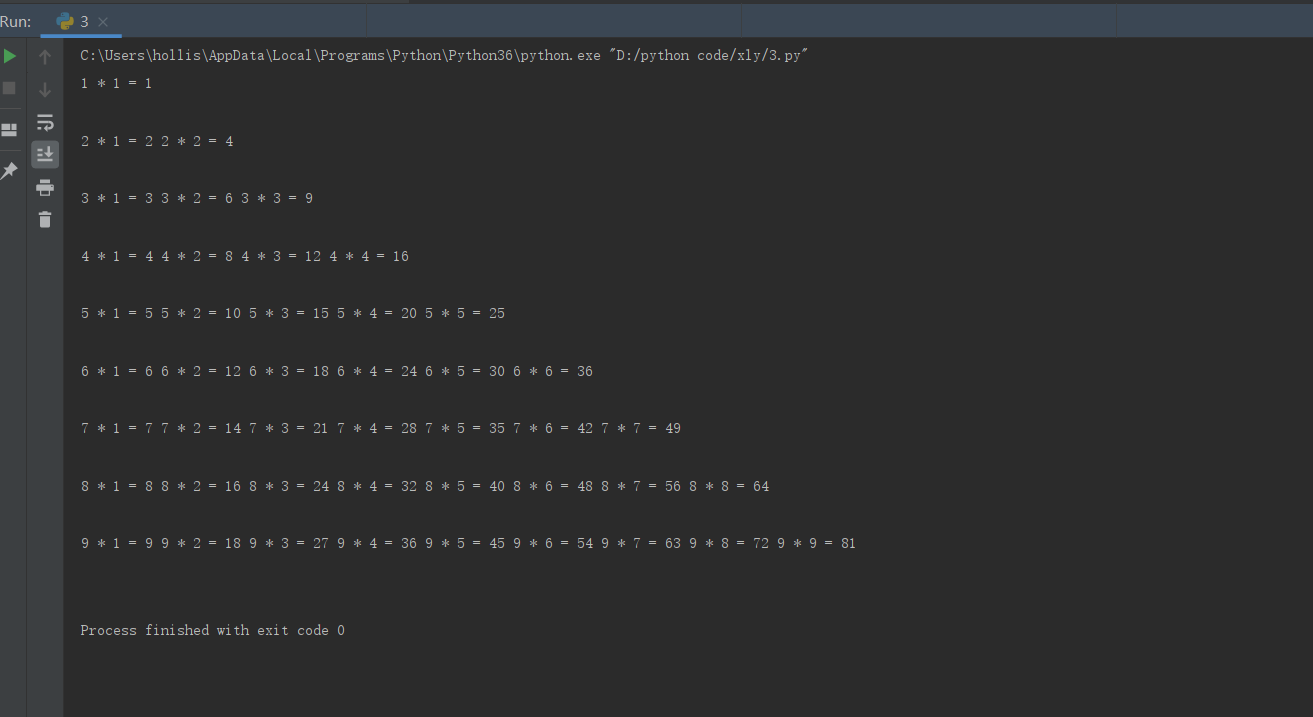
1、实现从1到100之间的奇数求和

print(sum([x for x in range(1,101) if x%2]))



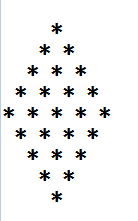
2、打印九九乘法表

for i in range(1,10):  
 for j in range(1,i+1):  
 print("%d \* %d = %d" % (i,j,i\*j),end=' ')  
 print('\n')

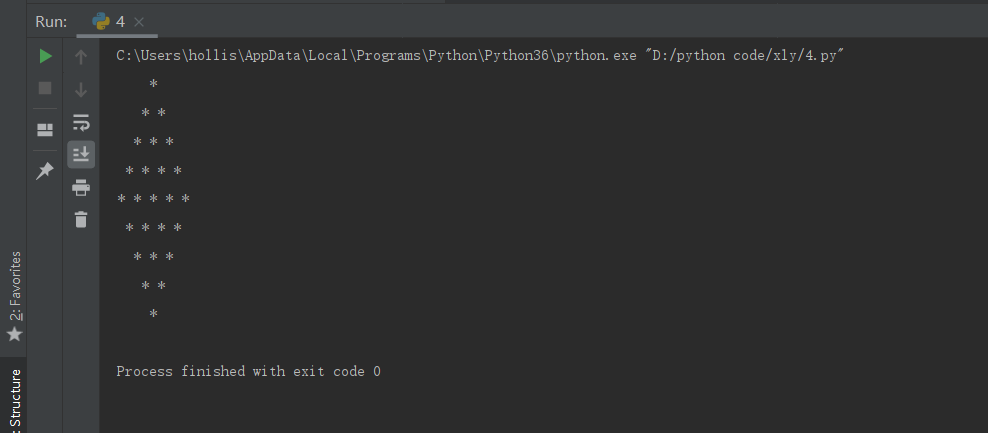


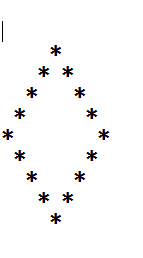
3、打印如下图形：

（1）

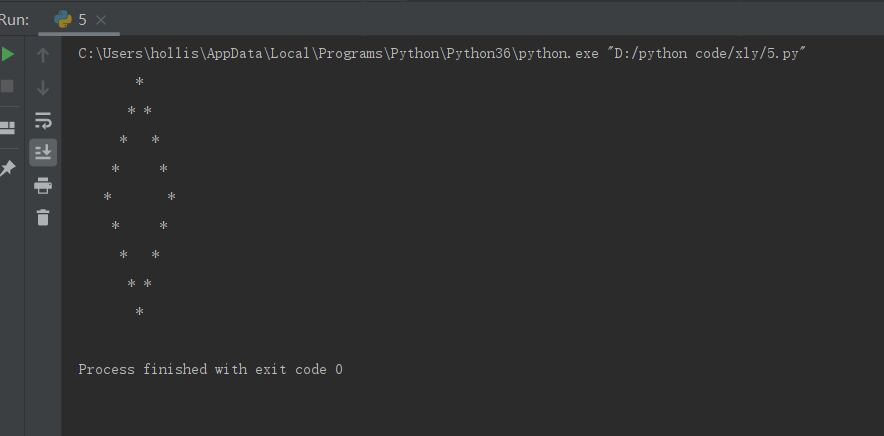


# line,char=input("line chat(use space to separate)").split()  
# line=int(line)  
# char=char+' '  
line=5  
char='\* '  
for i in range(1,line+1):  
 print((char\*i).center(len(char)\*line))  
for i in range(line-1,0,-1):  
 print((char\*i).center(len(char)\*line))





# line,char=input("line chat(use space to separate)：").split()  
# line=int(line)  
# l=len((char+' ')\*((line-1)\*2))  
  
line=5  
char='\*'  
l=len('\* '\*8) #最长行的长度  
  
for i in range(1,line+1):  
 if i==1:  
 print(char.center(l))  
 else:  
 tmp=char+' '\*((i-2)\*2+1)+char  
 print(tmp.center(l))  
  
for i in range(line-1,0,-1):  
 if i==1:  
 print(char.center(l))  
 else:  
 tmp=char+' '\*((i-2)\*2+1)+char  
 print(tmp.center(l))



4、统计一个整数对应的二进制数的1的个数。输入一个整数（可正可负）， 输出该整数的二进制包含1的个数

s=int(input("输入整数"))  
bin\_s=bin(s)  
  
if s>=0:  
 #正数补码=正数原码  
 num=bin\_s.count('1')  
else:  
 #对于负数，假设其为y，有  
 #[y]补码-->包括符号位在内全部取反（此时已为正数补码）+1-->[-y]补码=[-y]原码(计组)  
  
 num=64-bin(-s-1).count('1') #64位，-s-1为原负数补码取反   
  
print("%d 的2进制中1的个数为%d" %(s,num))

解法2

i = int(input("输入一个数:"))

cnt = 0

while i:

    i = i & (i - 1)

    cnt += 1

print(cnt)

5、有101个整数，其中有50个数出现了两次，1个数出现了一次， 找出出现了一次的那个数。

list = [1, 1, 2, 2, 3, 6, 6, 8, 3, 9, 8]

res = 0

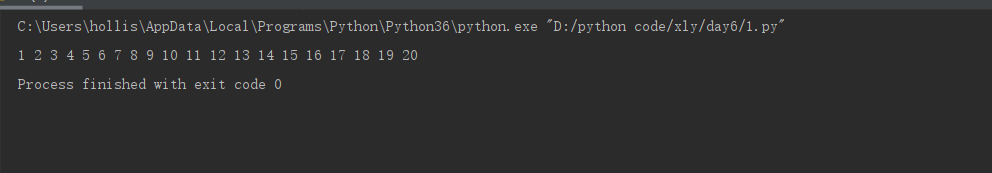
for i in list:

    res = res ^ i

print(res)

6 写一个简单的for循环，从1打印到20，横着打为1排

for i in range(1,21):  
 print(i,end=' ')

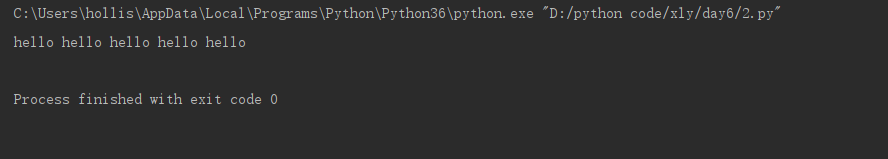


7 写一个say\_hello函数打印多次hello并给该函数加备注（具体打印几次依靠传递的参数），然后调用say\_hello，同时学会快速查看函数备注快捷键，及如何跳转到函数实现快捷键

快速查看函数快捷键：crtl按住，鼠标移动到函数调用处（但不点击）

函数实现处跳转快捷键：crtl+B或crtl+鼠标点击函数名称

def say\_hello(times):  
 *"""  
 重复打印指定次数的hello* ***:param*** *times: 打印次数* ***:return****: None  
 """* print("hello "\*times)  
  
  
  
say\_hello(5)



8 写一个模块（命名不要用中文），模块里写3个打印函数，然后另外一个py文件调用该模块，并调用对应模块的函数，同时用一下下面操作

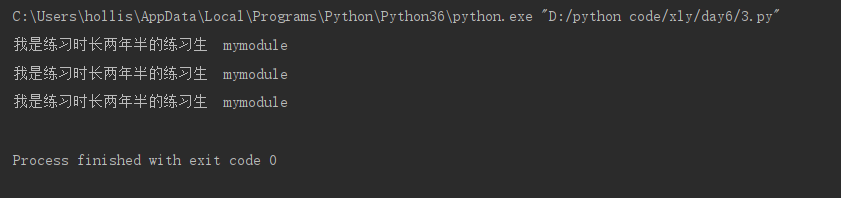
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 wd5.print\_line() *# 调用函数*

#mymodule 脚本文件

def print\_line():  
 print("我是练习时长两年半的练习生 ",\_\_name\_\_)  
  
print\_line()  
  
print\_line()  
  
if \_\_name\_\_=='\_\_main\_\_':  
 print\_line()

#导入自定义模块

import mymodule as mm  
  
mm.print\_line()



9 编写代码理解

不可变类型，内存中的数据不允许被修改：

数字类型 int, bool, float, complex, long(2.x)

字符串 str

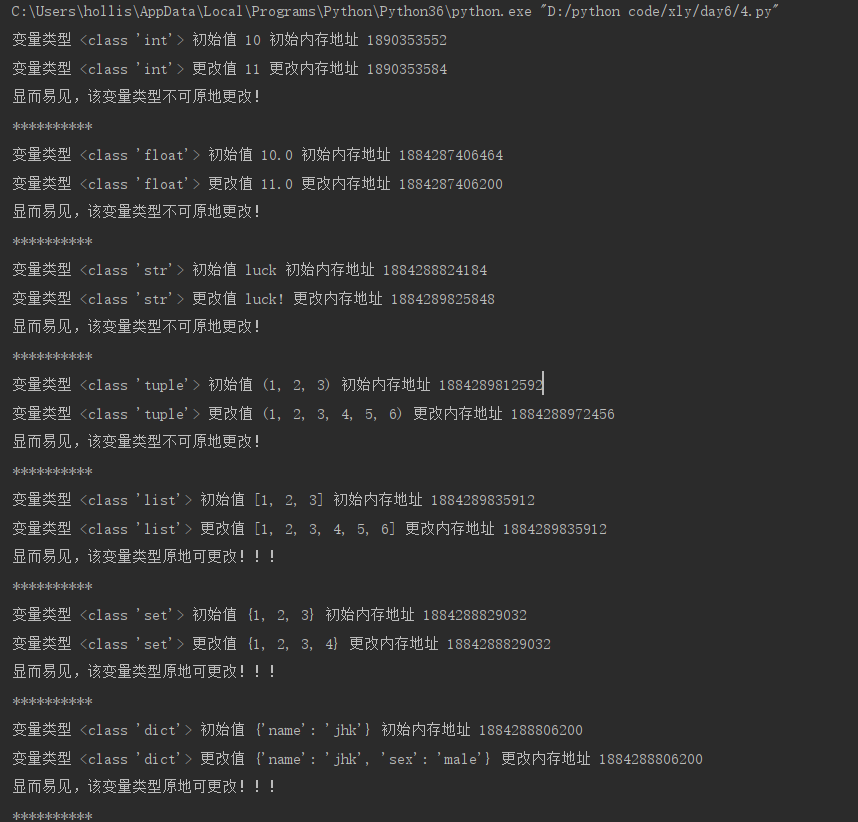
元组 tuple

可变类型，内存中的数据可以被修改：

列表 **list**

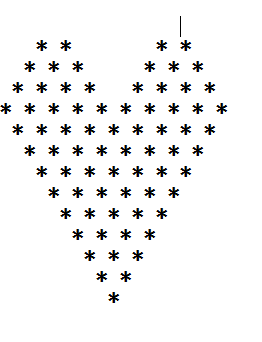
字典 **dict**

var\_type=[10,10.0,"luck",(1,2,3),[1,2,3],set([1,2,3]),{"name":'jhk'}]  
  
for i in var\_type:  
 t=type(i)  
  
 add1=id(i)  
 print("变量类型",t,"初始值",i,"初始内存地址",add1)  
  
 if isinstance(i,int): i+=1  
 elif isinstance(i,float):i+=1.0  
 elif isinstance(i,str):i+="!"  
 elif isinstance(i, tuple):i+=(4,5,6)  
 elif isinstance(i, list):i+=[4,5,6]  
 elif isinstance(i, set):i.add(4)  
 elif isinstance(i, dict):i.update({'sex':'male'})  
  
 add2=id(i)  
 print("变量类型", t, "更改值", i, "更改内存地址", id(i))  
  
 if add1==add2:  
 print("显而易见，该变量类型原地可更改！！！")  
 else:  
 print("显而易见，该变量类型不可原地更改！")  
 print('\*'\*10)

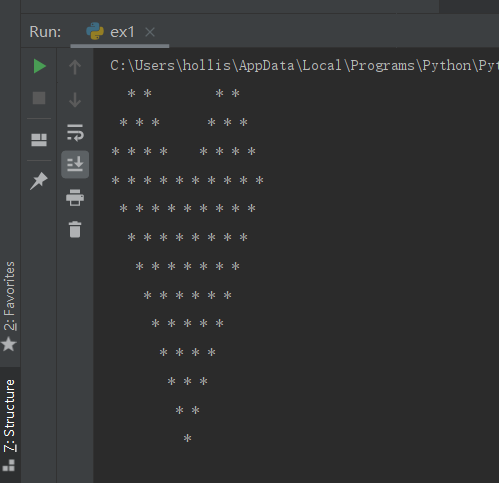


**难度作业**

1. 打印如下图形：



# max=int(input("how much are your love?(int only)"))  
  
max=10 #最大行  
l=len('\* '\*max)  
char='\* '  
  
#上半部分  
first=(max//2-1)  
uppatterns=[]  
for i in range(1,max):  
 char\_num=first-(i-1)  
 if char\_num<=1:break  
  
 tmp=(char\_num\*char+' '\*(3+(i-1)\*2)+char\_num\*char).center(l)  
 uppatterns.append(tmp)  
  
#逆序输出  
for i in range(len(uppatterns)-1,-1,-1):  
 print(uppatterns[i])  
  
  
#倒三角  
for i in range(max,0,-1):  
 print(('\* '\*i).center(l))



有101个整数，其中有50个数出现了两次，1个数出现了一次， 找出出现了一次的那个数。（大家使用7个数即可）

list = [1, 1, 2, 2, 3, 6, 6, 8, 3, 9, 8,10]

res = 0

res1 = 0

res2 = 0

for i in list:

    res = res ^ i

res &= -res

for i in list:

    if res & i:

        res1 ^= i

    else:

        res2 ^= i

print("%d %d" % (res1, res2))