```python
#习题册51页61题
for 行数 in range(1,6):
    print('*'*5)
```

控制台

```
*****
*****
*****
*****
*****
程序运行结束
```

```python
#习题册51页62题
for i in range(1,4):
    for j in range(4,7):
        print(i*j,end=',')
```

控制台

```
4,5,6,8,10,12,12,15,18,程序运行结束
```

```python
#习题册51页63题
list1=[1,2,3]
list2=[4,5,6]
L=[]
for i in range(3):
    L.append(list1[i]+list2[i])
print(L)
```

控制台

```
[5, 7, 9]
程序运行结束
```

```
#习题册51页64题，前期铺垫
for n in range(1,6):
    print('*'*n)
#通过代码和控制台可以看出，
# 随着n增加，每行的*也增加，
#这是字符串'*'和整数n相乘的效果
```

控制台

```
*
**
***
****
*****
程序运行结束
```

```
空格数=4
for n in range(1,6):
    print(' '*空格数+'*'*n)
    空格数-=1
#在控制台上，如果每行*的左边加入空格，
#这一行的*就会向右偏移

```

控制台

```
    *
   **
  ***
 ****
*****
程序运行结束
```

```
空格数=1
for n in range(1,3):
    print(' '*空格数+'*'*n)
    空格数-=1
#如果只是调整空格数和行数n，是无法在控制台上显示正三角形的
#因为若下一行的两个*之间没有空格，上一行的*就无法处于下一行的两个*中间

```

控制台

```
 *
**
程序运行结束
```

```
368 #习题册51页64题，前期铺垫三
369 空格数=1
370 for n in range(1,3):
371     print(' '*空格数+'* '*n)#必须把*和一个空格作为一个整体去乘n
372     空格数-=1
373 #才能使上一行的*处于下一行的*中间
374
```

控制台

```
 *
* *
程序运行结束
```

```
368 #习题册51页64题，前期铺垫三
369 空格数=4
370 for n in range(1,6):
371     print(' '*空格数+'* '*n)
372     空格数-=1
373 #调整空格数和行数n，可以做出5行正三角形的*图
374
```

控制台

```
    *
   * *
  * * *
 * * * *
* * * * *
程序运行结束
```

```python
#习题册51页64题，前期铺垫四
空格数=4
for n in range(1,6):
    print(' '*空格数+'* '*n)
    空格数-=1
    print()
    print()
    print()
    print()
#如果在输出完一行*号之后再输出4个换行
#看起来正三角形的*图被拉高了
```

控制台

```
    *


   * *


  * * *


 * * * *


* * * * *


程序运行结束
```

```python
#习题册51页64题
空格数=4
for n in range(1,6):
    print(' '*空格数+'* '*n)
    空格数-=1
    print('\n'*4)#'\n'就是换行的意思
#'\n'*4就是4个换行的意思
#运行效果和上图控制台一样
```

```
377 #习题册52页65题，铺垫知识:
378 matrix=[[1,2,3],[4,5,6],[7,8,9]]
379 #二维列表matrix的每个元素都是一维列表
380 for i in matrix:
381     print('matrix的元素',i)
382
```

控制台

```
matrix的元素 [1, 2, 3]
matrix的元素 [4, 5, 6]
matrix的元素 [7, 8, 9]
程序运行结束
```

```
377 #习题册52页65题，铺垫知识:
378 matrix=[[1,2,3],[4,5,6],[7,8,9]]
379 #二维列表matrix的每个元素都是一维列表
380 #每个一维列表的元素是一个整数
381 for i in matrix:#先遍历2维列表的元素，得到1维列表i
382     for j in i:#得到一维列表i，然后在i里面遍历每个整数元素j
383         print(j)
384
```

控制台

```
1
2
3
4
5
6
7
8
9
程序运行结束
```

```python
#习题册52页65题:
matrix=[[1,2,3],[4,5,6],[7,8,9]]
和=0
for i in matrix:
    for j in i:
        和+=j
print(和)

```

控制台

```
45
程序运行结束
```

```python

#习题册53页66题:
for m in range(1,6):#m是行号，从1到5，有5行
    for n in range(1,m+1):#n是列号，每行的列号从1输出到行号为止
        print(n,end=' ')#每输出一个列号，输出一个空格结尾
    print()#在控制台上，每次输出完一行列号，输出一个换行
    #print()里面什么都不写，默认就是输出一个换行

```

控制台

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
程序运行结束
```

```python

#习题册53页67题，做法1:
li=[1,2,3]
for i in li:
    for j in li:
        print(j,end=',')
    print()

```

控制台

```
1,2,3,
1,2,3,
1,2,3,
程序运行结束
```

```python
#习题册53页67题，做法2:
li=[1,2,3]
for i in li:
    for j in li:
        print(i,end=',')
    print()

```

控制台

```
1,1,1,
2,2,2,
3,3,3,
程序运行结束
```

```python
#习题册53页68题，铺垫:
#本题所讨论的组合为2个元素的组合
#实际上还可以讨论3个元素或4个元素的组合，等其他数量元素的组合
#并且本题所讨论的组合没有重复元素
#实际上还可以讨论有重复元素的组合

```

控制台

```
程序运行结束
```

```python
#习题册53页68题，铺垫:
#对于(1,'a',2,'b',True,False)这样的数据，用()合并起来叫做元组
#元组没有什么添加、修改、删除的方法
#和列表不同，元组在运行时不能添加、修改、删除；列表在运行时可以被添加、修改、删除。
```

```python
#习题册53页68题，铺垫:
# 对于列表[1,2,3,4]，其中有4个元素，选2个元素作组合:
#①先选元素1，再分别选2、3、4，和1作组合
#②选元素2，再分别选3、4，和2作组合
#③选元素3，再选4，和3作组合
```

```python
#习题册53页68题，铺垫:
# 对于列表[1,2,3,4]:
#元素的索引分别是0、1、2、3
#若元素1，则意味着选中了列表[0]
#分别选元素2、3、4，则意味着分别选中了列表[1]、列表[2]、列表[3]
#如果用i表示单个索引，用range(4)表示全部索引，
#则可以用for i in range(4):    表示i每次取到的索引
```

```
#习题册53页68题，铺垫：
# 对于列表[1,2,3,4]：
#根据前面提到的作两个元素组合的思路：①先选元素1，再分别选它后面的元素
#②再选2，再分别选它后面的元素
#③再选3，再选它后面的元素
#可以用代码 for i in range(3):
#                列表[i]
#表示先选了列表的元素1、2、3
#进一步用代码for i in range(3):
#                 for j in range(i+1,4):
#                     列表[i],列表[j]
#表示先选了列表的元素1、2、3，再分别选它们后面的元素
#i表示组合中第一个元素的索引，j表示组合中第2个元素的索引
#j比i大，从i+1开始，到整个列表的最后
```

```
431 #习题册53页68题，铺垫：
432 L=[1,2,3,4]
433 for i in range(3):
434     for j in range(i+1,4):
435         print(L[i],L[j])
436 #通过代码和控制台对比，总结一下2个元素组合的做法
437 #组合中的第一个元素的索引i从列表的第一到列表的倒数第二
438 #组合中第二个元素的索引j从i+1开始，到列表的最后
439
```

控制台

```
1 2
1 3
1 4
2 3
2 4
3 4
程序运行结束
```

```
442 #习题册53页68题
443 L=[1,2,3,4]
444 组合=[]
445 for i in range(3):
446     for j in range(i+1,4):
447         元组=(L[i],L[j])
448         组合.append(元组)
449 print(组合)
450
451
```

控制台

```
[(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]
程序运行结束
```

```
451 习题册54页69题，铺垫
452 #每一行的*数都是奇数
453 #第1行有1个*
454 #第2行有3个*
455 #第3行有5个*
456 #第4行有7个*
457 # 从上到下，*数递增的规律为：*数==行数*2-1
458 #如果仅仅将行数从1列举到4，*数可以通过行数的算式表示
459 for 行数 in range(1,5):
460     print('*'*(行数*2-1))
461
462
```

控制台

```
*
***
*****
*******
程序运行结束
```

```
461
462 #如果将第1行的*的左边补充3个空格，则第1行达到居中效果
463 #如果将第2行的*的左边补充2个空格，则第2行达到居中效果
464 #如果将第3行的*的左边补充1个空格，则第3行达到居中效果
465 #如果将第4行的*的左边补充0个空格，则第4行达到居中效果
466 #总结：如果总行数为4，空格的个数==总行数-当前第几行数
467
```

控制台

```
*
***
*****
*******
程序运行结束
```

```
469 #习题册54页69题，铺垫
470 总行数=4
471 for 行数 in range(1,5):
472     空格数=总行数-行数
473     字符串=' '*空格数+'*'*(行数*2-1)
474     print(字符串)
475
```

控制台

```
   *
  ***
 *****
*******
程序运行结束
```

```
#习题册54页69题，，铺垫
#对于从第4行到第7行，*数从上到下递减
#因为从上到下的前4行已经讨论出正确的规律
#为了不破坏已有成果，讨论从第5行向下
#第5行，5个*
#第6行，3个*
#第7行，1个*
#对于行号数和*数的关系：
#15-这一行的行号数*2==每行的*数
#即15-2*5==5
#即15-2*6==3
#即15-2*7==1
```

```
#习题册54页69题,铺垫
#然后再讨论从第5行向下，行号数和每行左边空格数的关系
#第5行，1个空格
#第6行，2个空格
#第7行，3个空格
#对于行号数和空格数的关系：
#这一行的行号数-4==每行的左边的空格数

#习题册53页69题，铺垫
# 总行数=4
# for 行数 in range(1,5):
#     空格数=总行数-行数
#     字符串=' '*空格数+'*'*(行数*2-1)
#     print(字符串)
#为了继承第1到第4行的控制台效果
#并保持两套思路的独立性，可以重新列举行数，并重新计算空格数和*数
for 行数 in range(5,8):
    空格数=行数-4
    星星数=15-行数*2
    字符串=' '*空格数+'*'*星星数
    print(字符串)
```

控制台

```
 *****
  ***
   *
程序运行结束
```

```
#习题册54页69题，铺垫
#把从上往下*数递增的代码和*数递减的代码合并起来
总行数=4
for 行数 in range(1,5):
    空格数=总行数-行数
    字符串=' '*空格数+'*'*(行数*2-1)
    print(字符串)
for 行数 in range(5,8):
    空格数=行数-4
    星星数=15-行数*2
    字符串=' '*空格数+'*'*星星数
    print(字符串)
```

控制台

```
   *
  ***
 *****
*******
 *****
  ***
   *
程序运行结束
```

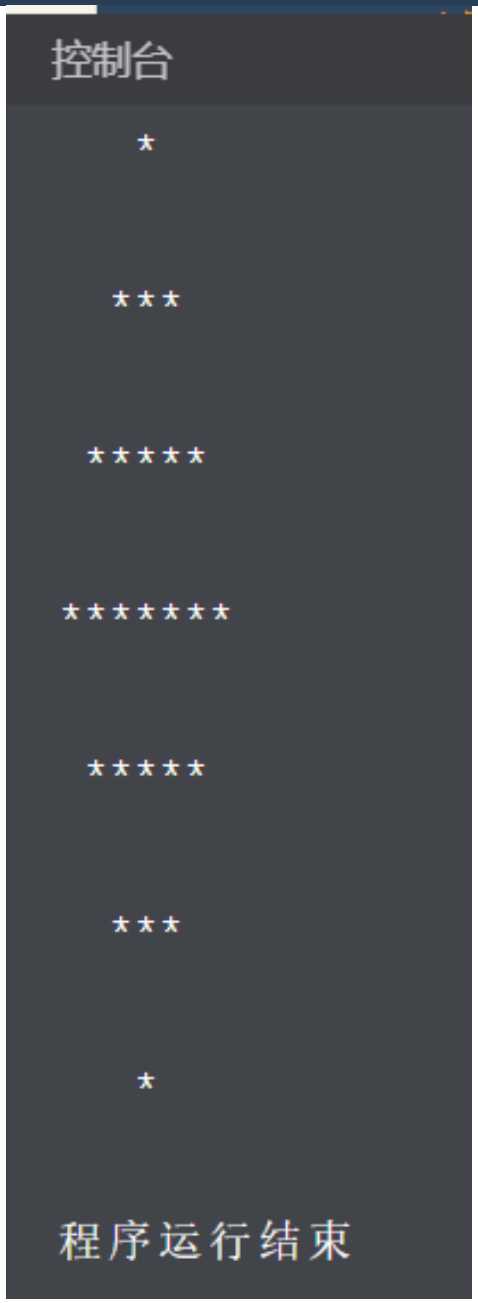
```
#习题册54页69题，去掉“总行数”这个变量（它只是整数4）
# 然后“星星数”和“空格数”都通过“行数”计算
#得到风格统一的两段代码
for 行数 in range(1,5):
    空格数=4-行数
    星星数=行数*2-1
    字符串=' '*空格数+'*'*星星数
    print(字符串)
for 行数 in range(5,8):
    空格数=行数-4
    星星数=15-行数*2
    字符串=' '*空格数+'*'*星星数
    print(字符串)

```

控制台

```
   *
  ***
 *****
*******
 *****
  ***
   *
程序运行结束
```

```
#习题册54页69题，如果在每次生成的字符串后面补上2个换行
#可以得到拉高的菱形
for 行数 in range(1,5):
    空格数=4-行数
    星星数=行数*2-1
    字符串=' '*空格数+'*'*星星数+'\n'*2
    print(字符串)
for 行数 in range(5,8):
    空格数=行数-4
    星星数=15-行数*2
    字符串=' '*空格数+'*'*星星数+'\n'*2
    print(字符串)
```



```
习题册54页69题，解法2
#观察菱形得到规律：总行数为7，每行星星数最多为7
#每行左边的空格数==(7-这行的星星数)//2
#从上到下，每行的星星数按1、3、5、7、5、3、1规律变化
#可以去掉行数这个变量，仅使用星星数的规律来做这道题
for 星星数 in range(1,8,2):
    空格数=(7-星星数)//2
    字符串=' '*空格数+'*'*星星数+'\n'*2
    print(字符串)
for 星星数 in range(5,0,-2):
    空格数=(7-星星数)//2
    字符串=' '*空格数+'*'*星星数+'\n'*2
    print(字符串)
```

控制台

```
   *


  ***


 *****


*******


 *****


  ***


   *

程序运行结束
```

```python
#习题册54页69题，解法3，
#如果行数从1列到7，星星数的递增和递减根据行数的大小来判断
#即行数<=5时，星星数=行数*2-1
#否则，星星数==15-行数*2
for 行数 in range(1,8):
    if 行数<=4:
        星星数=行数*2-1
    else:
        星星数=15-2*行数
    空格数=(7-星星数)//2
    字符串=' '*空格数+'*'*星星数+'\n'*2
    print(字符串)
```

控制台

```
   *


  ***


 *****


*******


 *****


  ***


   *

程序运行结束
```

```
520
521 #习题册54页70题，铺垫：搞清楚“不同位置元素的乘积”“不包括自身相乘”
522
523 #假设只有2个元素的列表[1,2]，其“不同位置元素的乘积”，只有1x2
524
525 #假设只有3个元素的列表[1,2,3]，其“不同位置元素的乘积”，
526 #有1x2，1x3，2x3，还有1x2x3
527
528 #假设只有4个元素的列表[1,2,3,4]，其“不同位置元素的乘积”，
529 #有1x2，1x3，1x4，2x3，2x4，3x4——【2个元素的组合】
530 #还有1x2x3，1x2x4，1x3x4，2x3x4，——【3个元素的组合】
531 #还有1x2x3x4————————————————【4个元素的组合】
532
533
```

```
538 #习题册54页70题，循环嵌套解法：
539 nums=[1,2,3,4]
540 L=[]
541 #步骤1，把2个元素的组合的乘积放进列表L
542 for i in range(3):
543     for j in range(i+1,4):
544         乘积=nums[i]*nums[j]
545         L.append(乘积)
546 #步骤2，把3个元素的组合的乘积放进列表L
547 for i in range(2):
548     for j in range(i+1,3):
549         for k in range(j+1,4):
550             乘积=nums[i]*nums[j]*nums[k]
551             L.append(乘积)
552 #步骤3，把4个元素的组合的乘积放进列表L
553 for i in range(1):
554     for j in range(i+1,2):
555         for k in range(j+1,3):
556             for p in range(k+1,4):
557                 乘积=nums[i]*nums[j]*nums[k]*nums[p]
558                 L.append(乘积)
559 print(L)
```

控制台

```
[2, 3, 4, 6, 8, 12, 6, 8, 12, 24, 24]
程序运行结束
```

```
562 #习题册54页70题，利用标准模块（虽然简单，但不合题意，题意是要用嵌套循环解法）：
563 from itertools import combinations#从迭代工具模块导入组合函数
564 nums=[1,2,3,4]
565 c=combinations(nums,2)#用组合函数求nums列表中2个元素的组合
566 print(c)#从控制台上看出，求出的组合是一个封装的，无法看出内部结构的对象
```

控制台

```
<itertools.combinations object at 0x0000025E3D845D00>
程序运行结束
```

```
563 from itertools import combinations
564 nums=[1,2,3,4]
565 c=combinations(nums,2)
566 for i in c:#遍历组合对象，输出每个组合
567     print(i)
```

控制台

```
(1, 2)
(1, 3)
(1, 4)
(2, 3)
(2, 4)
(3, 4)
程序运行结束
```

```
563 from itertools import combinations
564 nums=[1,2,3,4]
565 c=combinations(nums,3)
566 for i in c:#输出3个元素的组合
567     print(i)
```

控制台

```
(1, 2, 3)
(1, 2, 4)
(1, 3, 4)
(2, 3, 4)
程序运行结束
```

```
563 from itertools import combinations
564 nums=[1,2,3,4]
565 c=combinations(nums,4)
566 for i in c:#输出4个元素的组合
567     print(i)
568
```

控制台

```
(1, 2, 3, 4)
程序运行结束
```

```
562 #习题册54页70题，利用标准模块解法：
563 from itertools import combinations
564 nums=[1,2,3,4]
565 L=[]
566 for 组合数 in range(2,5):#让组合数从2列举到4
567     c=combinations(nums,组合数)#根据每种组合数，生成一个组合对象
568     for i in c:#遍历组合对象，得到组合i
569         乘积=1
570         for j in i:
571             乘积=乘积*j#计算组合i中所有元素的乘积
572         L.append(乘积)#将乘积添加进列表
573 print(L)#也用到了嵌套循环，应该也是符合题意的
574
```

控制台

```
[2, 3, 4, 6, 8, 12, 6, 8, 12, 24, 24]
程序运行结束
```

```
575 #习题册55页71题
576 for i in range(1,11):
577     if i==5:
578         print('循环非正常结束，i进行到',i)
579         break
580     else:
581         print('循环正常进行，i进行到',i)
582
583
584
585
```

控制台

```
循环正常进行，i进行到 1
循环正常进行，i进行到 2
循环正常进行，i进行到 3
循环正常进行，i进行到 4
循环非正常结束，i进行到 5
程序运行结束
```

```python
#习题册55页72题
li=[1,2,3,4,5,6,7,8,9,10]
for i in li:
    if i%2==0:
        continue
    print(i,end=',')
```

控制台

```
1,3,5,7,9,程序运行结束
```

```python
#习题册55页73题
计数器=1
重复判断的循环次数=0
while True:
    if 计数器==8:
        break
    计数器+=1
    重复判断的循环次数+=1
print(计数器,重复判断的循环次数)
```

控制台

```
8 7
程序运行结束
```

```python
#习题册56页74题
string="Hello,World"
for i in string:
    if i==',':
        continue
    print(i)
```

控制台

```
H
e
l
l
o
W
o
r
l
d
程序运行结束
```

```python
#习题册56页75题
for i in range(1,16):
    if i%3==0:
        pass
    else:
        print(i)

```

控制台

```
1
2
4
5
7
8
10
11
13
14
程序运行结束
```

```python
#习题册56页76题
nums=[1,2,3,4,5,6,7,8,9,10]
i=0
while i<len(nums):
    if nums[i]>5:
        break
    print(nums[i])
    i+=1


```

控制台

```
1
2
3
4
5
程序运行结束
```

```python
#习题册57页77题
for i in range(1,11):
    if i==7:
        continue
    print(i)
    if i!=10:
        pass
    else:
        print('循环结束，未遇到中断情况（除continue外）')
```

控制台

```
1
2
3
4
5
6
8
9
10
循环结束，未遇到中断情况（除continue外）
程序运行结束
```

```python
#习题册57页78题
dic={'a':1,'b':2,'c':3,'d':4,'e':5}
for i in dic:
    if i=='c':
        break
    print(i,dic[i])
```

控制台

```
a 1
b 2
程序运行结束
```

```python
#习题册57页79题
和=0
个数=0
while True:
    n=float(input('请输入数字，（负数退出）：'))
    if n<0:
        break
    和+=n
    个数+=1
print('所有输入的正数的平均值为：',和/个数)
```

控制台

```
请输入数字，（负数退出）：1
请输入数字，（负数退出）：2
请输入数字，（负数退出）：-1
所有输入的正数的平均值为： 1.5
程序运行结束
```