

1. 《编程导论》的《程序：春游坐车问题3》(A) 这个程序有问题。请问你问题在哪里？假如 $student=89$, $seat=30$, 会如何？(B) 请改正此程序，让其面面俱到。

将你的代码与运行结果截图贴在此处。

(A)当数值为 $student=89$, $seat=30$ 时，会出现一辆车的座位量大于车辆的最大座位量

(B)修改：

```
student = int(input())
# get the value, pay attention to the function int()
seat = 30
num = student // seat + int((student % seat) != 0)
bus = []
# create the list of bus to put students in it
bus += [0] * num
# initialize the list of bus
i = student
# protect student from being changed
while i > 0:
    bus[i % num] += 1
    i -= 1
# put student in the list in turn
if student % num == 0:
    print("%d bus(es). Each have %d student(s)" % (num, bus[0]))
else:
    print("%d bus(es). " % num)
    print("%d bus(es) has(have) %d student(s)" % (bus.count(max(bus)), max(bus)))
    print("%d bus(es) has(have) %d student(s)" % (bus.count(min(bus)), min(bus)))
```

Shell:

Case0:

```
70
3 bus(es).
1 bus(es) has(have) 24 student(s)
2 bus(es) has(have) 23 student(s)

Process finished with exit code 0
```

Case1:

```
89
3 bus(es).
2 bus(es) has(have) 30 student(s)
1 bus(es) has(have) 29 student(s)

Process finished with exit code 0
```

Case2:

```
182
4 bus(es).
2 bus(es) has(have) 26 student(s)
2 bus(es) has(have) 25 student(s)

Process finished with exit code 0
```

Case3:

```
1200  
40 bus(es). Each have 30 student(s)  
  
Process finished with exit code 0
```

这个代码几乎是按照自己从《计算机导论》所学的知识重新写的。这个程序的泛用性更广了。

2.用Python 完成《编程导论》练习题 1.4.4, 1.4.5, 1.4.6, 1.4.7, 请自己设定一个列表L, 执行程序来验证你的结果。将你的代码与运行结果截图贴在此处。

```
La = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ]
Lb = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ]
# given lists
L1_1 = La[: len(La)//2]
L1_2 = La[len(La) // 2:]
print("1.4.4 output:", L1_1, L1_2)
# Ex1.4.1
L2_1 = Lb[: len(Lb)//2]
L2_2 = Lb[len(Lb) // 2:]
print("1.4.5 output:", L2_1, L2_2)
# Ex1.4.5
L3_1 = La[: len(La)//3]
L3_2 = La[len(La)//3: 2*len(La)//3]
L3_3 = La[2*len(La)//3:]
print("1.4.6 output:", L3_1, L3_2, L3_3)
# Ex1.4.6
L4 = La[::-1]
print("1.4.7 output:", L4)
# Ex1.4.7
```

Shell:

```
1.4.4 output: [0, 1, 2, 3, 4] [5, 6, 7, 8, 9]
1.4.5 output: [10, 11, 12, 13, 14] [15, 16, 17, 18, 19, 20]
1.4.6 output: [0, 1, 2] [3, 4, 5] [6, 7, 8, 9]
1.4.7 output: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

Process finished with exit code 0

3. 《编程导论》的《程序：解二维线性方程组3》。假设方程式1是 $3x-2y=1$ 。给出不同的方程式2，使得结果是独特解，无解，或无限多解。

将你的代码与运行结果截图贴在此处。

```
A = [[3, -2], [0,0]]
B = [1, 0]
# given linear system of equations
A[1][0], A[1][1] = map(int, input().split())
B[1] = int(input())
# get the value
if A[0][0] == 0:
    y = B[0]/A[0][1]
    x = (B[1]-A[1][1]*y)/A[1][0]
# a situation
elif A[1][0] == 0:
    y = B[1]/A[1][1]
    x = (B[0]-A[0][1]*y)/A[0][0]
# the situation similar to the up
else:
    b = A[0][1]*A[1][0] - A[1][1] * A[0][0]
    c = B[0] * A[1][0] - B[1] * A[0][0]
    if b == 0 and c == 0:
        print("Infinite Solution!")
    elif b == 0:
        print("No solution!")
    else:
        y = c / b
        x = (B[0]-A[0][1]*y)/A[0][0]
    print("x = %.3f, y = %.3f" % (x, y))
```

Shell:

Case0:

```
4 5
-1
x = 0.130, y = -0.304
```

Process finished with exit code 0

Case1:

```
-3 2
-1
Infinite Solution!
```

Process finished with exit code 0

Case2:

```
0 -4
1
No solution!
```

Process finished with exit code 0

4.完成并试验下列练习题的Python 程序: 练习题 1.5.3, 1.5.4, 1.5.5;
将你的代码与运行结果截图贴在此处。

1.5.3:

```
k = 10
for i in range(0, k):
    for j in range(1, 6):
        print(j, end=' ')
    print('\n')
```

Shell:

12345

12345

12345

12345

12345

12345

12345

12345

12345

12345

Process finished with exit code 0

1.5.4:

```
n = 15
k = 10
for i in range(0, k):
    for j in range(1, n+1):
        print(j, end=' ')
    print('\n')
```

Shell:

5.完成并试验下列练习题的 Python 程序: 仿造练习题 1.5.5, 打印出一个倒三角形的图形, 也就是, 第一行是 $1\dots n$, 第二行是 $1\dots n-1$, 最后一行是 1。
将你的代码与运行结果截图贴在此处。

```
n = 15
for i in range(n, 0, -1):
    for j in range(1, i+1):
        print(j, end=' ')
    print('\n')
```

Shell:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 2 3 4 5 6 7 8 9 10 11 12 13
1 2 3 4 5 6 7 8 9 10 11 12
1 2 3 4 5 6 7 8 9 10 11
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Process finished with exit code 0

6.完成并执行 Python 程序: (A) 输入任意四个不同数 a, b, c, d , 输出它们的最小数和第二小的数。例如输入 $a, b, c, d = 8, 2, 1, 10$; 输出的值为 1, 2。(B) 请设计程序使输入为一个长度未知的列表, 输出其中的最小数和第二小的数。完成程序后用大小在 $[1, 100]$ 内的随机数生成一个长度为 50 的列表验证程序 (尝试用一行代码生成)。将你的代码与运行结果截图贴在此处。

(A):

```
a, b, c, d = map(int, input().split())
l = [a, b, c, d]
l_min = min(l)
while l_min == min(l):
    l.remove(min(l))
# use while to remove all the maximum
l_sec_min = min(l)
print(l_min, l_sec_min)
```

Shell:

```
3 4 1 6
1 3
Process finished with exit code 0
```

(B):

```
a = list(map(int, input().split()))
a_min = min(a)
while a_min == min(a):
    a.remove(min(a))
# use while to remove all the minimum
a_sec_min = min(a)
print(a_min, a_sec_min)
```

生成随机数组的方法:

```
import numpy
a = list(numpy.random.randint(1, 101, 50))
```

生成随机数组后, 就不会将有

```
a = list(map(int, input().split()))
```

这句话

Shell:

Case0:

```
2 2 2 4 5 6 5 4 5
2 4
Process finished with exit code 0
```

Case1:

```
[23, 34, 21, 25, 51, 74, 26, 26, 54, 63, 68, 71, 35, 78, 67, 14, 41, 36, 79, 19, 67, 64, 15, 16, 69, 74, 17, 21, 5, 31, 1, 68, 7, 66, 49, 21, 66, 47, 12, 92, 66, 98, 76, 98, 29, 85, 91, 65, 18, 97]
1 5
Process finished with exit code 0
```


7.判断一个点是否在矩形内部(含边界) 平面直角坐标系中有一点 $A(ax, ay)$, 以点 A 的纵坐标为基准, 分别向上、向下移动高度为 h 的距离作两条平行于 x 轴的直线; 以点 A 的横坐标为基准, 分别向左、向右移动长度为 w 的距离作两条平行于 y 轴的直线。取四条直线围成的矩形(如下图所示)。现在有另一个点 $C(cx, cy)$, 判断点 C 是否在该矩形内部(包含边界)。如果在, 程序输出 Yes, 否则输出 No。(假设 $ax=3, ay=3, h=2, w=2.5, cx=4, cy=1$ 。), 请试试看不同的数值, 例如 $cx=5.5, cy=0.5$ 。

将你的代码与运行结果截图贴在此处。

```
ax, ay = 3, 3
w, h = 2.5, 2
cx, cy = map(float, input().split())
if abs(ax-cx) <= w and abs(ay-cy) <= h:
    print('YES')
else:
    print('NO')
```

Shell:

Case0:

```
4 1
YES

Process finished with exit code 0
```

Case1:

```
5.5 0.5
NO

Process finished with exit code 0
```