

# Python (acwing)

## 1.顺序结构

```
1 print(type(123))
2 print(type(1.23))
3 print(type('acwing'))
```

1. PS:Python 自带高精度

2. " 与 '是等价的

3. 变量直接用

```
1 width = 20
2 height = 12.3
3 flag = True
4 a,b,c=1,'a',3.0
5 print(a,b,c)
```

```
1 print(a+b)  #计算两个数的和
2 print("my name" + 'is mmm')
```

```
1 a=10
2 b=20
3 print(a+b)
4 print(a*b)
5 print(a/b)
6 print(a//b) #两个除号代表整除,向下取整
7 print(a%b)
8 print(a ** b) #2都三次方
9 c = a + b
10 print(c)
11 a += b #累加运算
12 #同理-=,*=,/=,//=,**=都可以简写
```

```
13 x = 2 + 3
14 y = (x + 1,0) * 5
15 print(x,y)
16 s = '123.0'
17 ss = float(s) #将字符串中的浮点数提取出来
18 b = int(s) #取整
19 s2 = "the valve is " + str(b) #将整数转化成字符串
```

```
1 a = int(input())
2 b = int(input())
3 print(a + b) #要将输入转化成字符串
4 s = input()
5 print(s.split()) #split将元素拆分
6 a,b=map(int, input().split()) #先用input读进来，再用split函数分成隔了空格的字符串，最后用int作用
7 print(a+b)
```

```
1 a,b=map(int, input().split()) #先用input读进来，再用split函数分成隔了空格的字符串，最后用int作用
2 print(a+b)
3 print(1,end = 'xx')
4 print(2,end = '')
5 print(3,end = ' ')
6 #使用print()函数可以输出多个变量，默认用空格隔开，且行末会默认添加回车。
7 #如果想自定义行末字符，可以添加end参数。
```

```
1 x = 12.34567
2 print('x = '+str(round(x,2)))
3 y = 12
4 print("x = %.1f" % x) #保留一位小数
5 print("x = %.3f"%(x * y)) #保留三位小数,先计算乘积，然后再计算这两个数的值
6 print("x = %f,y = %d"%(x,y)) #输出整数，%f默认保留六位
```

```
1 from math import sqrt # 引入开根号函数
2 x = 2
3 y = sqrt(x)
4 print(y)
```

# 作业

```
1 604.圆的面积
2 a = float(input())
3 b = a*a * 3.14159
4 print("A=%.4f"%b)
5
6 1.A+B
7 a,b = map(int,input().split())
8 print(a + b)
9
10 606.平均数1
11 a = float(input())
12 b = float(input())
13 a *= (3.5/11)
14 b *= (7.5/11)
15 print("MEDIA = %.5f"%(a + b))
16
17 609.工资
18 a = int(input())
19 b = int(input())
20 c = float(input())
21 print("NUMBER = %d"%a)
22 print("SALARY = U$ %.2f"%(b * c))
23
24 608.差
25 a = int(input())
26 b = int(input())
27 c = int(input())
28 d = int(input())
29 print("DIFERENCA = %d"%(a*b-c*d))
30
31 616. 两点间的距离
32 x1,y1=map(float,input().split())
33 x2,y2=map(float,input().split())
34 a = (x1 - x2) ** 2
35 b = (y1 - y2) ** 2
36 from math import sqrt
37 c = sqrt(a + b)
38 print("%.4f"%c)
39
40 653. 钞票
41 a = int(input())
42 a1 = a%100
43 a2 = a1%50
44 a3 = a2%20
```

```

45 a4 = a3%10
46 a5 = a4%5
47 a6 = a5%2
48 print(a)
49 print("%d nota(s) de R$ 100,00"%(a//100))
50 print("%d nota(s) de R$ 50,00"%(a1//50))
51 print("%d nota(s) de R$ 20,00"%(a2//20))
52 print("%d nota(s) de R$ 10,00"%(a3//10))
53 print("%d nota(s) de R$ 5,00"%(a4//5))
54 print("%d nota(s) de R$ 2,00"%(a5//2))
55 print("%d nota(s) de R$ 1,00"%(a6))
56
57 615. 油耗
58 a = int(input())
59 b = float(input())
60 print("%.3f km/l"%(a/b))
61
62 654. 时间转换
63 a = int(input())
64 b = a // 3600
65 c = (a%3600)//60
66 d = (a%3600)%60
67 print("%d:%d:%d"%(b,c,d))
68
69 605. 简单乘积
70 a = int(input())
71 b = int(input())
72 print("PROD = %d"%(a * b))
73
74 611. 简单计算
75 x,y,z = map(float,input().split())
76 a,b,c = map(float,input().split())
77 print("VALOR A PAGAR: R$ %.2f"%(y*z+b*c))
78
79 612. 球的体积
80 a = int(input())
81 pi = 3.14159
82 a **= 3
83 v = (4/3)*pi*a
84 print("VOLUME = %.3f"%v)
85
86 613. 面积
87 a,b,c=map(float,input().split())
88 print("TRIANGULO: %.3f"%(0.5*a*c))
89 print("CIRCULO: %.3f"%(3.14159*c*c))
90 print("TRAPEZIO: %.3f"%(0.5*(a+b)*c))
91 print("QUADRADO: %.3f"%(b**2))

```

```
92 print("RETANGULO: %.3f"%(b*a))
93
94 607. 平均数2
95 a = float(input())
96 b = float(input())
97 c = float(input())
98 a*=0.2
99 b*=0.3
100 c*=0.5
101 print("MEDIA = %.1f"%(a+b+c))
102
103 610. 工资和奖金
104 a = input()
105 b = float(input())
106 c = float(input())
107 d = b + 0.15 * c
108 print("TOTAL = R$ %.2f"%d)
109
110 614. 最大值
111 x,y,z=map(int,input().split())
112 c = (x+y+abs(x-y))/2
113 d = (c + z+abs(c-z))/2
114 print("%d eh o maior"%d)
115
116 617. 距离
117 a = int(input())
118 print("%d minutos"%(a/30*60))
119
120 618. 燃料消耗
121 t = int(input())
122 s = int(input())
123 s = (s * t) / 12
124 print("%.3f"%s)
125
126 656. 钞票和硬币
127 a = float(input())
128 a*=100
129 a1 = a%10000
130 a2 = a1%5000
131 a3 = a2%2000
132 a4 = a3%1000
133 a5 = a4%500
134 a6 = a5%200
135 a7 = a6%100
136 a8 = a7%50
137 a9 = a8%25
138 a10 = a9%10
```

```

139 a11 = a10%5
140 print("NOTAS:")
141 print("%d nota(s) de R$ 100.00"%(a//10000))
142 print("%d nota(s) de R$ 50.00"%(a1//5000))
143 print("%d nota(s) de R$ 20.00"%(a2//2000))
144 print("%d nota(s) de R$ 10.00"%(a3//1000))
145 print("%d nota(s) de R$ 5.00"%(a4//500))
146 print("%d nota(s) de R$ 2.00"%(a5//200))
147 print("MOEDAS:")
148 print("%d moeda(s) de R$ 1.00"%(a6//100))
149 print("%d moeda(s) de R$ 0.50"%(a7//50))
150 print("%d moeda(s) de R$ 0.25"%(a8//25))
151 print("%d moeda(s) de R$ 0.10"%(a9//10))
152 print("%d moeda(s) de R$ 0.05"%(a10//5))
153 print("%d moeda(s) de R$ 0.01"%(a11//1))
154
155 #float类型的数值(浮点数)有精度问题, 存储的是近似值。
156 #例如print(0.1 + 0.2)会得到0.30000000000000004
157
158 655. 天数转换
159 a = int(input())
160 print("%d ano(s)"%(a//365))
161 a%=365
162 print("%d mes(es)"%(a//30))
163 a%=30
164 print("%d dia(s)"%a)

```

## 2. 判断语句

### 笔记

```

1 '''
2 1. 基本if-else语句
3 当条件成立时, 执行某些语句; 否则执行另一些语句。
4 注意:
5
6 if和else后需要加上冒号:
7 if语句的代码块需要缩进统一长度, 规范写法是缩进4个空格。
8 a = int(input())
9 '''
10 # 正确写法:
11 if a > 5:
12     print("%d is big!" % a)
13     print("%d + 1 = %d" % (a, a + 1))

```

```
14 else:
15     print("%d is small!" % a)
16     print("%d - 1 = %d" % (a, a - 1))
17
18
19 #当只有一条语句时，可以写在同一行，但这样写不规范，只要了解即可：
20
21 a = int(input())
22
23 if a > 5: print("%d is big!" % a)
24 else: print("%d is small!" % a)
25
26 #练习：输入一个整数，输出这个数的绝对值
27 a = int(input())
28 if a < 0:
29     a = -a
30 print("%d"%a)
31
32 #练习：输入两个整数，输出两个数中较大的那个
33 a,b=map(int,input().split())
34 if a>b:
35     print(a)
36 else
37     print(b)
38
39 #练习：输入三个整数，输出三个数中最大的那个
40 a,b,c=map(int,input().split())
41 if a > b:
42     if a > c:
43         print(a)
44     else:
45         print(c)
46 else:
47     if b > c:
48         print(b)
49     else:
50         print(c)
51 '''
52 2. 常用比较运算符
53 (1) 大于 >
54 (2) 小于 <
55 (3) 大于等于 >=
56 (4) 小于等于 <=
57 (5) 等于 ==
58 (6) 不等于 !=
59 '''
60
```

```
61 #ctrl + 方向键可以跳左右括号
62
63 '''
64 3. if-else连写
65 注意: else if要写成elif
66 '''
67
68
69 #练习:
70 #输入一个0到100之间的分数,
71 #如果大于等于85, 输出A;
72 #如果大于等于70并且小于85, 输出B;
73 #如果大于等于60并且小于70, 输出C;
74 #如果小于60, 输出 D;
75
76 a = int(input())
77 if a >= 85:
78     print('A')
79 elif a >= 70:
80     print('B')
81 elif a >= 60:
82     print('C')
83 else:
84     print('D')
85
86 # '''
87 # 练习:
88 #
89 # 1.判断闰年。闰年有两种情况:
90 # (1) 能被100整除时, 必须能被400整除;
91 # (2) 不能被100整除时, 被4整除即可。
92 # 输入一个年份, 如果是闰年输出yes, 否则输出no。
93 #
94 # '''
95
96 a = int(input())
97 if a % 100 == 0:
98     if a % 400 == 0:
99         print('yes')
100 elif a % 4 == 0:
101     print('yes')
102 else:
103     print('no')
104
105 '''
106 4. pass 语句
107 pass 语句不执行任何动作。语法上需要一个语句, 但程序毋需执行任何动作时, 可以使用该语句。
```



```
108
109 '''
110 x = int(input())
111 if x > 5:
112     pass
113 else:
114     print(x)
115
116 '''
117
118 5. 变量的作用域
119 if语句内部的变量，可以在语句外访问。
120
121 '''
122 a,b=map(int,input().split())
123 if a > b:
124     max = a
125 else:
126     max = b
127 print(max)
128
129 '''
130 二、条件表达式
131 (1) 与 and
132 (2) 或 or
133 (3) 非 not
134
135 注意：运算符优先级：not > and > or。
136 '''
137
138 #练习：用一条if语句，判断闰年
139 a = int(input())
140 if a%100!=0 and a%4==0 or a%400==0:
141     print('yes')
142 else
143     print('no')
144
145 '''
146
147 三、三元运算
148 类似于C++、Java中的问号表达式。例如：
149
150 '''
151 a,b=map(int,input().split())
152 max = a if a>b else b
153
154 '''
```

#### 155 四、match语句

156 python3.10开始新增了match语句。目前作业评测器的Python3采用旧版本，尚未支持match语法。

157

158 注意：

159

160 只有第一个匹配的模式会被执行。且跟C++、Java不同，匹配后只会执行当前模式，不会顺次执行后面的case。

161 可以用 | 表示匹配多个模式。

162 变量名 \_ 被作为 通配符 并必定会匹配成功。

163 如果没有 case 匹配成功，则不会执行任何分支。

164

165 '''

166 status = int(input())

167

168 match status:

169 case 400:

170 print("Bad request")

171 case 404:

172 print("Not found")

173 case 418 | 420 | 422:

174 print("I'm a teapot")

175 case \_:

176 print("Something's wrong with the internet")

177

178

179 '''

180 注意if和else语句后一定要加冒号。

181 判断语句内部的代码一定要记得缩进。

182 格式化字符串中想表示%时，需要写%%。

183 Python中交换两个变量，可以用：a, b = b, a。

184 Python中的比较运算符支持链式操作，这一点跟C++和Java等语言不同。例如，给三个数排序的代码可以这么写：

185 a, b, c = map(int, input().split())

186 x, y, z = a, b, c

187

188 if a >= b >= c:

189 print(c, b, a)

190 elif a >= c >= b:

191 print(b, c, a)

192 elif b >= a >= c:

193 print(c, a, b)

194 elif b >= c >= a:

195 print(a, c, b)

196 elif c >= a >= b:

197 print(b, a, c)

198 elif c >= b >= a:

199 print(a, b, c)

```
200
201 '''
```

## 作业

```
1 #665. 倍数
2 a,b=map(int,input().split())
3 if a % b == 0 or b % a == 0:
4     print('Sao Multiplos')
5 else:
6     print('Nao sao Multiplos')
7
8 #660. 零食
9 x,y = map(int,input().split())
10 if x == 1:
11     print("Total: R$ %.2f" % (y * 4.00))
12 elif x == 2:
13     print("Total: R$ %.2f" % (y * 4.50))
14 elif x == 3:
15     print("Total: R$ %.2f" % (y * 5.00))
16 elif x == 4:
17     print("Total: R$ %.2f" % (y * 2.00))
18 else:
19     print("Total: R$ %.2f" % (y * 1.50))
20
21 #669. 加薪
22 a = float(input())
23 if a <= 400.00:
24     print("Novo salario: %.2f"%(a*1.15))
25     print("Reajuste ganho: %.2f"%(0.15*a))
26     print("Em percentual: 15 %")
27 elif a <=800.00:
28     print("Novo salario: %.2f" % (a * 1.12))
29     print("Reajuste ganho: %.2f" % (0.12 * a))
30     print("Em percentual: 12 %")
31 elif a <=1200.00:
32     print("Novo salario: %.2f" % (a * 1.1))
33     print("Reajuste ganho: %.2f" % (0.1 * a))
34     print("Em percentual: 10 %")
35 elif a <=2000.00:
36     print("Novo salario: %.2f" % (a * 1.07))
37     print("Reajuste ganho: %.2f" % (0.07 * a))
38     print("Em percentual: 7 %")
39 else:
40     print("Novo salario: %.2f" % (a * 1.04))
```

```
41     print("Reajuste ganho: %.2f" % (0.04 * a))
42     print("Em percentual: 4 %")
43
44 #671.DDD
45 a = int(input())
46 if a == 61:
47     print("Brasilia")
48 elif a == 71:
49     print("Salvador")
50 elif a == 11:
51     print("Sao Paulo")
52 elif a == 21:
53     print("Rio de Janeiro")
54 elif a == 32:
55     print("Juiz de Fora")
56 elif a == 19:
57     print("Campinas")
58 elif a == 27:
59     print("Vitoria")
60 elif a == 31:
61     print("Belo Horizonte")
62 else:
63     print("DDD nao cadastrado")
64
65 #667. 游戏时间
66 a,b=map(int,input().split())
67 if b > a:
68     print("O JOGO DUROU %d HORA(S)"%(b - a))
69 elif a >= b:
70     print("O JOGO DUROU %d HORA(S)" % (b - a + 24))
71
72 #663. 简单排序
73 a,b,c=map(int,input().split())
74 x = a
75 y = b
76 z = c
77 if a < b:
78     if a > c:
79         a,c = c,a
80     if b > c:
81         b,c = c,b
82 else:
83     a,b=b,a
84     if a > c:
85         a,c = c,a
86     if b > c:
87         b,c = c,b
```

```
88
89 print(a)
90 print(b)
91 print(c)
92 print(' ')
93 print(x)
94 print(y)
95 print(z)
96
97 #657. 选择练习1
98 a,b,c,d=map(int,input().split())
99 if b > c and d > a and c + d > a + b and c > 0 and d > 0 and a % 2 == 0:
100     print("Valores aceitos")
101 else:
102     print("Valores nao aceitos")
103
104 #664. 三角形
105 a,b,c=map(float,input().split())
106 if a+b>c and b+c > a and c + a > b:
107     print("Perimetro = %.1f"%(a+b+c))
108 else:
109     print("Area = %.1f"%(0.5*(a+b)*c))
110
111 #659. 区间
112 a = float(input())
113 if a < 0 or a > 100:
114     print("Fora de intervalo")
115 elif 25>=a>=0:
116     print("Intervalo [0,25]")
117 elif 25<a<=50:
118     print("Intervalo (25,50]")
119 elif 50<a<=75:
120     print("Intervalo (50,75]")
121 elif 75<a<=100:
122     print("Intervalo (75,100]")
123
124 #662. 点的坐标
125 a,b=map(float,input().split())
126 if a == 0 and b == 0:
127     print("Origem")
128 elif a > 0 and b > 0:
129     print("Q1")
130 elif a < 0 and b > 0:
131     print("Q2")
132 elif a < 0 and b < 0:
133     print("Q3")
134 elif a > 0 and b < 0:
```

```

135     print("Q4")
136 elif a == 0:
137     print("Eixo Y")
138 elif b == 0:
139     print("Eixo X")
140
141 #672. 税
142 a = float(input())
143 if a<=2000:
144     print("Isento")
145 elif 3000>=a>2000:
146     print("R$ %.2f"%((a-2000)*0.08))
147 elif 4500>=a>3000:
148     print("R$ %.2f"%((a-3000)*0.18+80))
149 elif a>4500:
150     print("R$ %.2f"%((a-4500)*0.28+350))
151
152 #668. 游戏时间2
153 a,b,c,d=map(int,input().split())
154 if b > d:
155     d += 60
156     c -= 1
157     if c < a:
158         c += 24
159 elif a == c and b == d:
160     c+=24
161 else:
162     if c < a:
163         c +=24
164
165 x = c - a
166 y = d - b
167 print("O JOGO DUROU %d HORA(S) E %d MINUTO(S)"%(x,y))
168
169 #658. 一元二次方程公式
170 a,b,c=map(float,input().split())
171 if (b ** 2) - 4 * a * c < 0:
172     print("Impossivel calcular")
173 elif a == 0:
174     print("Impossivel calcular")
175 else :
176     print("R1 = %.5f"%((-b+((b ** 2) - 4 * a * c)**0.5)*0.5*1/a))
177     print("R2 = %.5f" % ((-b - ((b ** 2) - 4 * a * c) ** 0.5) * 0.5 * 1 / a))
178
179 #666. 三角形类型
180 a, b, c = map(float, input().split())
181

```

```

182 if a < b:
183     a, b = b, a
184 if a < c:
185     a, c = c, a
186 if b < c:
187     b, c = c, b
188
189 if a >= b + c:
190     print("NAO FORMA TRIANGULO")
191 else:
192     if a * a == b * b + c * c:
193         print("TRIANGULO RETANGULO")
194     if a * a > b * b + c * c:
195         print("TRIANGULO OBTUSANGULO")
196     if a * a < b * b + c * c:
197         print("TRIANGULO ACUTANGULO")
198     if a == b == c: # 等价于 a == b and b == c
199         print("TRIANGULO EQUILATERO")
200     elif a == b or b == c:
201         print("TRIANGULO ISOSCELES")
202
203 #661. 平均数3
204 a, b, c, d = map(float, input().split())
205 x = (a * 2 + b * 3 + c * 4 + d) / 10
206 print("Media: %.1f" % x)
207
208 if x >= 7:
209     print("Aluno aprovado.")
210 elif x < 5:
211     print("Aluno reprovado.")
212 else:
213     print("Aluno em exame.")
214     y = float(input())
215     print("Nota do exame: %.1f" % y)
216     z = (x + y) / 2
217     if z >= 5:
218         print("Aluno aprovado.")
219     else:
220         print("Aluno reprovado.")
221     print("Media final: %.1f" % z)
222

```

### 3.循环语句

#### 笔记

```
1 '''
2
3 一、while循环
4 可以简单理解为循环版的if语句。if语句是判断一次，如果条件成立，则执行后面的语句；while是每
   次判断，如果条件成立，则执行循环体中的语句，否则停止。
5
6 注意，类似于if语句，while语句也需要满足：
7
8 while后需要加上冒号：
9 while语句的代码块需要缩进统一长度，规范写法是缩进4个空格。
10
11 '''
12 i = 0
13 while i < 10:
14     print(i)
15     i += 1
16
17 #练习：求1~100中所有数的立方和。
18 s = 0
19 i = 1
20 while i <= 100:
21     s += i ** 3
22     i += 1
23 print(s)
24
25 #练习：求斐波那契数列的第n项。  $f(1) = 1, f(2) = 1, f(n) = f(n - 1) + f(n - 2)$ 
26 n = int(input())
27 a, b, i = 1, 1, 1
28 while i < n
29     a, b = b, a + b
30     i += 1
31 print(a)
32
33 '''
34 二、for循环
35 for循环语句用来按顺序枚举range、字符串等数据类型中的元素。类似于while和if语句，for语句同
   样需要满足冒号和缩进的要求。
36
37 注意：本节课重在学习for循环，而非学习列表、元组、集合、字典等复杂数据类型，所以for语句与这
   些数据类型配合的使用技巧会放到下一章中展开。
38
39 1. 遍历字符串
40 for语句可以遍历字符串中的每个字符。
41
42 '''
43
```



```

44 for c in 'python':
45     print(c,end=' ')
46
47 '''
48 2. 遍历range
49 range()函数可以生成等差数列，可以接收1个、2个或者3个整数参数：
50
51 接收1个整数参数时：range(x)会按顺序返回 0,1,2,3,...x-1这个数列。
52 接收2个整数参数时：range(x, y)会按顺序返回 x,x+1,x+2,...,y-1这个数列。
53 接收3个整数参数时：range(x, y, z)分为两种情况：
54 z > 0时，按顺序返回 x,x+z,x+2z,x+3z,.....这个数列中小于 y的所有数。
55 z < 0时，按顺序返回 x,x+z,x+2z,x+3z,.....这个数列中大于 y的所有数。
56
57 '''
58 for i in range(5):
59     print(i,end='')
60 print()
61
62 for i in range(3,8):
63     print(i,end='')
64 print()
65
66 for i in range(1,10,2):
67     print(i, end='')
68 print()
69
70 for i in range(0,-10,-2):
71     print(i, end='')
72 print()
73
74 #练习：求1~100中所有数的立方和。
75 s = 0
76 for i in range(1,101):
77     s += i **3
78 print(s)
79
80 #练习：求斐波那契数列的第n项。  $f(1) = 1, f(2) = 1, f(n) = f(n - 1) + f(n - 2)$ 。
81 n = int(input())
82 a,b=1,1
83 for i in range(n-1):
84     a,b = b,a+b
85 print(a)
86
87 '''
88 三、跳转语句
89 1. break
90 可以提前从最近的一层循环中退出，一般与if语句搭配。

```

91 例题：判断一个大于1的整数是否是质数。

92 '''

93 n = int(input())

94 is\_prime = True

95 for i in range(2,n):

96 if n % i == 0:

97 is\_prime = False

98 break

99

100 if is\_prime:

101 print("yes")

102 else:

103 print("no")

104

105 '''

106 2. continue

107 可以跳过当前这次迭代后面的语句，并继续下一次迭代。作用与if语句类似。

108 例题：求 1~100中所有偶数的和。

109

110 '''

111 s = 0

112 for i in range(1,101):

113 if i % 2 == 1:

114 continue

115 s += i

116 print(s)

117

118 '''

119 四、循环中的else子句和pass语句

120 本节内容用得不多，了解即可。

121

122 1. 循环中的else子句

123 for或while循环可以包括else子句，会在循环结束后执行。

124 不过如果循环是被break结束的，那么else子句就不会执行了。

125 '''

126 #例题：判断一个大于1的整数是否是质数。

127 n = int(input())

128

129 for x in range(2,n):

130 if n % x == 0:

131 print("%d = %d \* %d"%(n,x,n//x))

132 break

133 else:

134 print("%d is a prime number"%n)

135

136 '''

137 2. pass语句

```
138 类似于if语句，当for或者while循环里不想写任何代码时，可以写上pass语句，这个语句不执行任何
    动作。
139
140 '''
141
142 for i in range(10): # 不进行任何操作
143     pass
144
145 while True: # 死循环
146     pass
147
148 '''
149 五、多层循环
150 while和for循环内的代码块中也可以包含循环语句。
151
152 '''
153 #例题：将1~100打印到一个10 * 10的矩阵中：
154 k = 0
155 for i in range(0, 10):
156     for j in range(0, 10):
157         print(k, end=' ')
158         k += 1
159     print() # 输出回车
160
161
162 #练习：打印 1~100中的所有质数。
163 for i in range(2,101):
164     is_prime = True
165     for j in range(2,i):
166         if i % j == 0:
167             is_prime = False
168             break
169     if is_prime:
170         print(i)
171
172 '''
173 六、作业题易错点
174 注意多层for循环中所使用的循环变量尽量使用不同的变量名，防止相互干扰。
175 Python3中求一个数的绝对值，可以使用abs()函数。
176 '''
```

## 作业

```
1 #708. 偶数
2 for i in range(3,102,2):
```

```

3     print(i-1)
4
5  #709. 奇数
6  n = int(input())
7  for i in range(0,n,2):
8      print(i+1)
9
10 #712. 正数
11 x = 0
12 for i in range(0,6):
13     n = float(input())
14     if n <= 0:
15         continue
16     x += 1
17 print("%d positive numbers"%x)
18
19 # #721. 递增序列
20 n = 1
21 while n != 0:
22     n = int(input())
23     if n == 0:
24         break
25     for i in range(1,n+1):
26         print(i ,end = ' ')
27     print()
28
29 #723. PUM
30 u = 1
31 m,n = map(int,input().split())
32 for i in range(1,m+1):
33     for j in range(1,n):
34         print(u,end = ' ')
35         u+=1
36     print('PUM')
37     u+=1

```

## 四、列表

### 笔记

```

1  '''
2  列表是一种常见的变量类型，可以存储一系列元素。元素的类型可以不同，不过大多数情况下类型是相同的。
3

```

```
4 1. 列表的初始化
5 列表需要用方括号括起来，中间的元素用逗号隔开，最后一个元素后可以多加一个逗号。
6
7 1.1 直接初始化
8 '''
9 a = [] # 定义一个空列表
10 b = [1, 2] # 含有2个整数的列表
11 c = [6, "python", 3.14] # 含有3个不同类型元素的列表
12 print(a, b, c)
13 '''
14 1.2 通过append函数初始化
15 append()函数可以在列表末尾添加一个元素。
16 '''
17 a = []
18 for i in range(5):
19     a.append(0)
20 print(a)
21
22 n = 4
23 b = []
24 for i in range(n):
25     b.append(i*i)
26 print(b)
27
28 '''
29 1.3 通过for语句初始化列表长度和每个位置的数值
30 '''
31 a = [0 for i in range(5)]
32 print(a)
33
34 n = 4
35 b = [i * i for i in range(n)]
36 print(b)
37
38 '''
39 2. 访问列表元素
40 可以通过下标读/写列表中的元素，下标从0开始，也可以是负数，负数下标表示的是除以列表长度的余数对应的位置。
41
42 负数下标相当于将列表首尾相接，然后从0往前数。
43 如果列表长度是 n，那么下标只能取 -n~-1之间的整数，超出范围会报错。
44
45 '''
46 a = [0, 1, 2]
47 print(a[0], a[1], a[2])
48
49 a[0] = 3
```

```

50 print(a)
51 print(a[-1]) # 由于-1 % 3 = 2, 所以表示的是a[2]
52
53 '''
54 3. 使用循环语句遍历列表
55 可以通过下标访问, 例如:
56
57 '''
58
59 a = [2,1,3,4,3]
60 for i in range(5):
61     print(a[i],end='')
62
63 #可以通过for ... in ...直接遍历, 例如:
64 a = [2,3,45,5]
65 for x in a:
66     print(x,end='')
67
68
69 #练习题: 假设斐波那契数列的前两项分别是 a0=0,a1=1求 an
70 n = int(input())
71 a = [0 for i in range(n + 1)]
72 a[0]=0
73 a[1]=1
74 for i in range(2,n+1):
75     a[i] = a[i -1]+a[i-2]
76 print(a[n])
77
78 '''
79 4. 列表的切片操作
80 列表的切片操作会返回一个新列表。用法:
81
82 a[begin:end] 会返回包含a[begin], a[begin + 1], ..., a[end - 1]的列表。
83 省略begin时, begin的默认值是0。
84 省略end时, end的默认值是列表长度。
85 如果begin或end是负数, 表示的是除以列表长度后的余数。
86
87 '''
88
89 a=[0,1,2,3,4]
90 print(a[1:4])
91 print(a[1:])
92 print(a[:4])
93 print(a[:])
94 print(a[-4:-1])
95
96 '''

```

```
97 切片也支持写操作：
98  '''
99  a = [0, 1, 2, 3, 4]
100
101  a[1:4] = [2, 3]
102  print(a)  # 输出[0, 2, 3, 4]
103
104  '''
105  5. 列表的复制
106  列表跟其他变量类型不同，如果用等号复制列表，那么原列表和新列表是同一个列表。
107  对新列表修改，会修改原列表的值。例如：
108  '''
109  a = [0,1,2,3,4]
110  b = a
111  b[0]=5
112
113  '''
114  使用切片，可以得到一个原列表的浅拷贝。此时再修改新列表，原列表就不会发生变化了。例如：
115  '''
116  a = [0,1,2,3,4]
117  b = a[:]
118  b[0]=5
119  print(a,b)
120
121  '''
122  6. 列表的运算
123  列表的加法可以将两个列表拼接起来，得到一个新列表。
124  列表乘以一个整数，可以将若干个自身拼接起来，得到一个新列表。
125  例如：
126  '''
127
128  a = [0,1,2]
129  b = [3,4]
130  c = a + b
131  print(c) # 输出[0, 1, 2, 3, 4]
132  d = a * 3
133  print(d) # 输出[0, 1, 2, 0, 1, 2, 0, 1, 2]
134  a += [3]
135  print(a) # 输出[0, 1, 2, 3]
136
137  '''
138  8. 嵌套列表
139  列表中的元素也可以是列表，例如：
140
141  '''
142  matrix = [
143      [0,1,2],
```

```

144     [3,4,5],
145     [6,7,8],# 最后一个元素后可以添加逗号
146 ]
147 matrix[1][0] = 99# 修改嵌套列表中元素的值
148 print(matrix[1][0])# 读取嵌套列表中元素的值
149
150 for i in range(len(matrix)):
151     for j in range(len(matrix[i])):
152         print(matrix[i][j],end = ' ')
153     print()
154
155 for row in matrix: # 循环打印整个矩阵
156     for x in row:
157         print(x, end=' ')
158     print() # 在行末输出回车

```

```

159
160 '''

```

嵌套列表的常用初始化方式有两种。

```

162

```

采用append()函数初始化：

```

164 '''

```

```

165 matrix = []
166 for i in range(3):
167     row = []
168     for j in range(3):
169         row.append(i * 3 + j)
170     matrix.append(row)
171
172 print(matrix)
173
174
175 '''
176 采用for语句直接初始化：
177 '''
178 matrix = [[i*3+j for j in range(3)]for i in range(3)]
179 print(matrix)

```

```

180

```

```

181 '''

```

## 9. 作业题扩展内容

当需要读入一行整数时，可以用：a = list(map(int, input().split()))。类似地，当需要读入一行浮点数时，可以用a = list(map(float, input().split()))：

min(a, b)可以求两个数的最小值。类似得min(a, b, c)可以求三个数的最小值，以此类推。同样地max()可以求最大值。

a[::-1]可以返回一个逆序的新列表

```

186 '''

```



# 作业

```
1 #737. 数组替换
2 a = []
3 for i in range(10):
4     n = int(input())
5     a.append(n)
6     if n <= 0:
7         a[i] = 1
8 for i in range(10):
9     print("x[%d] = %d" % (i, a[i]))
10
11 #738. 数组填充
12 v = int(input())
13
14 a = [v * 2 ** i for i in range(10)]
15
16 for i in range(10):
17     print("N[%d] = %d" % (i, a[i]))
18
19 743. 数组中的行
20 r = int(input())
21 t = input()
22
23 a = []
24 for i in range(12):
25     row = list(map(float, input().split()))
26     a.append(row)
27
28 s = 0
29 for i in range(12):
30     s += a[r][i]
31
32 if t == "M":
33     s /= 12
34
35 print("%.1f" % s)
36
37 745. 数组的右上半部分
38 t = input()
39 a = []
40 for i in range(12):
41     row = list(map(float, input().split()))
42     a.append(row)
43
44 s = 0
```

```
45 c = 0
46 for i in range(11):
47     for j in range(i + 1, 12):
48         s += a[i][j]
49         c += 1
50
51 if t == "M":
52     s /= c
53
54 print("%.1f" % s)
55
56 749. 数组的上方区域
57 t = input()
58 a = []
59 for i in range(12):
60     row = list(map(float, input().split()))
61     a.append(row)
62
63 s = 0
64 c = 0
65 for i in range(5):
66     for j in range(i + 1, 11 - i):
67         s += a[i][j]
68         c += 1
69
70 if t == "M":
71     s /= c
72
73 print("%.1f" % s)
74
75 753. 平方矩阵 I
76 while True:
77     n = int(input())
78     if n == 0:
79         break
80
81     for i in range(n):
82         for j in range(n):
83             x = min(i, n - 1 - i, j, n - 1 - j)
84             print(x + 1, end=' ')
85         print()
86
87     print()
88
89 739. 数组选择
90 a = []
91 for i in range(100):
```

```
92     x = float(input())
93     a.append(x)
94
95 for i in range(len(a)):
96     if a[i] <= 10:
97         print("A[%d] = %.1f" % (i, a[i]))
98
99 740. 数组变换
100 a = []
101 for i in range(20):
102     x = int(input())
103     a.append(x)
104
105 a.reverse()
106
107 for i in range(len(a)):
108     print("N[%d] = %d" % (i, a[i]))
109
110 741. 斐波那契数列
111 f = [0 for i in range(61)]
112 f[0] = 0
113 f[1] = 1
114 for i in range(2, 61):
115     f[i] = f[i - 1] + f[i - 2]
116
117 T = int(input())
118 for i in range(T):
119     n = int(input())
120     print("Fib(%d) = %d" % (n, f[n]))
121
122 742. 最小数和它的位置
123 n = int(input())
124 a = list(map(int, input().split()))
125
126 k = 0
127 for i in range(1, n):
128     if a[i] < a[k]:
129         k = i
130
131 print("Minimum value:", a[k])
132 print("Position:", k)
133
134 744. 数组中的列
135 c = int(input())
136 t = input()
137 a = []
138 for i in range(12):
```

```
139     row = list(map(float, input().split()))
140     a.append(row)
141
142     s = 0
143     for i in range(12):
144         s += a[i][c]
145
146     if t == "M":
147         s /= 12
148
149     print("%.1f" % s)
150
151 747. 数组的左上半部分
152 t = input()
153 a = []
154 for i in range(12):
155     row = list(map(float, input().split()))
156     a.append(row)
157
158 s, c = 0, 0
159 for i in range(12):
160     for j in range(0, 11 - i):
161         s += a[i][j]
162         c += 1
163
164 if t == "M":
165     s /= c
166
167 print("%.1f" % s)
168
169 748. 数组的右下半部分
170 t = input()
171 a = []
172 for i in range(12):
173     row = list(map(float, input().split()))
174     a.append(row)
175
176 s, c = 0, 0
177 for i in range(0, 12):
178     for j in range(12 - i, 12):
179         s += a[i][j]
180         c += 1
181
182 if t == "M":
183     s /= c
184
185 print("%.1f" % s)
```

```

186
187 746. 数组的左下半部分
188 t = input()
189 a = []
190 for i in range(12):
191     row = list(map(float, input().split()))
192     a.append(row)
193
194 s, c = 0, 0
195 for i in range(12):
196     for j in range(0, i):
197         s += a[i][j]
198         c += 1
199
200 if t == "M":
201     s /= c
202
203 print("%.1f" % s)

```

## 五、字符串

### 笔记

```

1 '''
2 1. 字符与整数的联系—ASCII码
3 每个常用字符都对应一个-128 ~ 127的数字，二者之间可以相互转化。注意：目前负数没有与之对应的字符。
4
5 ord()函数可以求一个字符的ASCII码。注意输入是一个字符，而不是字符串。
6 chr()函数可以将一个ASCII码转化成对应的字符。
7
8 '''
9 c = 'a'
10 print(ord(c))
11
12 a = 66
13 print(chr(a))
14
15 '''
16 常用ASCII值：'A' - 'Z'是65 ~ 90，'a' - 'z'是97 - 122，0 - 9是 48 - 57。
17
18 注意：虽然字符可以跟整数相互转化，
19 但在Python中，字符不能参与数值运算，这一点跟C++、Java等语言是不同的。
20

```

## 21 2. 字符串常量的写法

22 在Python中，字符串既可以用单引号来表示，也可以用双引号来表示，二者完全相同。

23 这一点跟C++、Java等编程语言是不同的，

24 在这些编程语言中，用单引号来表示字符，用双引号来表示字符串。

25 '''

26 a = 'hello world'

27 print(a)

28

29 b = "hello wrold"

30 print(b)

31

32 #两个或多个字符串常量并排写，会被自动合并，例如：

33 a = "MY" "name " "is yxc"

34

35 '''

36 一个字符串如果包含多行，

37 可以采用"""..."""或者''' '''的初始化方式，字符串中将自动包含回车字符，例如：

38

39 '''

40 a = """Usage: thingy [OPTIONS]

41 -h Display this usage message

42 -H hostname Hostname to connect to"""

43 print(a)

44

45 '''

## 46 3. 表示特殊字符——转义

47 当想在字符串中表示特殊字符时，一般可以在字符前加反斜杠\。常见需要转义的字符有：

48

49 转义字符      含义    ASCII码（十进制）

50 \n    回车    10

51 \\    代表一个反斜杠\    92

52 \"    表示一个双引号    34

53 \'    表示一个单引号    39

54

55 '''

56 print("My name is:\n\"yxc!\")

57

58 #另外，如果想输出单引号，也可以用双引号来表示，反之亦然。例如

59 print("My name is 'yxc!') # 输出: My name is 'yxc!'

60 print('My name is "yxc!') # 输出: My name is "yxc!'

61 '''

## 62 4. 访问字符串中的每个字符

63 可以通过下标读取字符串中的每个字符，下标从0开始，也可以是负数，负数下标表示的是除以字符串长度的余数对应的位置。

64

65 负数下标相当于将字符串首位相接，然后从0往前数。

66 如果字符串长度是 n，那么下标只能取 -n~-1之间的整数，超出范围会报错。

```
67
68 注意：字符串中的每个字符不能修改。
69 '''
70 a = "hello world"
71 print(a[0],ord(a[5]))
72 #a[2] = 'x' # 会报错，字符串不能修改
73 '''
74 5. 使用循环语句遍历字符串
75 可以通过下标访问，例如：
76 '''
77 s = "acwing"
78 for i in range(6):
79     print(s[i],end=' ')
80 print()
81 #可以通过for ... in ...直接遍历，例如：
82 for c in "python":
83     print(c, end=' ') # 注意c本身也是字符串类型
84 print() # 输出回车
85
86 '''
87 6. 字符串的切片操作
88 字符串的切片操作会返回一个新字符串。用法：
89
90 a[begin:end] 会返回包含a[begin], a[begin + 1], ..., a[end - 1]的字符串。
91 省略begin时，begin的默认值是0。
92 省略end时，end的默认值是字符串长度。
93 如果begin或end是负数，表示的是除以字符串长度后的余数。
94 例如：
95 '''
96
97 a = "ABCDE"
98 print(a[1:4]) # 输出BCD
99 print(a[1:]) # 输出BCDE
100 print(a[:4]) # 输出ABCD
101 print(a[:]) # 输出ABCDE
102 print(a[-4:-1]) # 等价于print(a[1:4])
103 #注意：字符串的切片不支持写操作。
104
105 '''
106 例如：
107
108 a = "ABCDE"
109 a[1:4] = "XY" # 会报错，字符串不能修改
110 '''
111 '''
112 7. 字符串的复制
113 跟列表不同，字符串的每次复制操作，都会得到一个全新的字符串。
```

```
114
115 在 Python 中，字符串是不可变的，这意味着你不能直接修改字符串中的某个字符。不过，你可以通过
    创建新的字符串来实现类似的效果。以下是一些常见的方法来“修改”字符串内容：
116
117 1. 使用切片和拼接
118 你可以通过切片和拼接的方式来构造新的字符串。例如，如果你想将字符串 "hello" 中的第三个字
    符 'l' 修改为 'p'，可以这样做：
119
120 '''
121 original_string = "hello"
122 new_string = original_string[:2] + 'p' + original_string[3:]
123 print(new_string) # 输出: heplo
124 '''
125
126 2. 使用 'join' 和列表
127 你可以将字符串转换为列表，修改列表中的元素，然后再将列表转换回字符串。例如：
128 '''
129
130 original_string = "hello"
131 char_list = list(original_string)
132 char_list[2] = 'p'
133 new_string = ''.join(char_list)
134 print(new_string) # 输出: heplo
135 '''
136
137 3. 使用字符串格式化
138 如果你只是想替换字符串中的某个特定子字符串，可以使用字符串的 'replace' 方法：
139 '''
140
141 original_string = "hello"
142 new_string = original_string.replace('l', 'p')
143 print(new_string) # 输出: heppo
144 '''
145
146 4. 使用正则表达式
147 如果你需要更复杂的字符串替换逻辑，可以使用正则表达式模块 're'：
148
149 '''
150 import re
151
152 original_string = "hello"
153 new_string = re.sub('l', 'p', original_string)
154 print(new_string) # 输出: heppo
155 '''
156
157 示例：修改字符串中的特定位置
```



```
158 假设你想将字符串 "hello" 中的第三个字符 'l' 修改为 'p'，可以使用上述任意一种方法。以下是使用切片和拼接的方法：
159 '''
160 original_string = "hello"
161 position = 2 # 第三个字符的位置（索引从0开始）
162 new_char = 'p'
163 '''
164 使用切片和拼接
165 '''
166 new_string = original_string[:position] + new_char + original_string[position
    + 1:]
167 print(new_string) # 输出: heppo
168 ```
169
170 '''
171 8. 字符串的运算
172 字符串的加法可以将两个字符串拼接起来，得到一个新字符串。
173 字符串乘以一个整数，可以将若干个自身拼接起来，得到一个新字符串。
174 字符串支持比较运算符，按字典序比较大小。即如果两个字符串相同，则表示相等；
175 否则找到两个字符串从左到右数第一个不一样的字符，哪个字符串的字符的ASCII码小，
176 哪个字符串的字典序就小；另外空字符比任何字符都小。
177 例如：
178 '''
179 a = "Hello "
180 b = "World"
181 c = a + b
182 print(c) # 输出Hello World
183
184 d = a * 3
185 print(d) # 输出Hello Hello Hello
186
187 e = a * 3 + "World"
188 print(e) # 输出Hello Hello Hello World
189
190 print(a <= b) # 按字典序比较大小，输出True
191 print("123" > "22") # 按字典序比较大小，输出False
192
193 '''
194 9. 字符串的常用操作
195 假设s是一个字符串，则：
196
197 len(s)返回字符串长度。
198 s.split(sep)返回一个字符串列表。如果给出了sep就按sep分隔；如果没给出，则会按空格分隔，但
    连续的空格会被视为单个分隔符，而且会忽略首尾的空白字符。
199 s.strip()将首尾的空白字符删除。
200 s.replace(old, new)将s中所有的old子串都改成new。
201 s.find("abc")查询某个子串在s中第一次出现的下标；如果不存在，则返回-1。
```

```

202 s.startswith(prefix)判断prefix是否为s的前缀。
203 s.endswith(suffix)判断suffix是否为s的后缀。
204 s.lower()将所有大写字母变成小写。
205 s.upper()将所有小写字母变成大写。
206 s.join(a)，a是一个字符串列表，这个函数返回将a中的字符用s作为分隔符拼接起来的结果。
207 注意：返回的所有字符串都是新字符串，原字符串不变。
208
209 例如：
210 '''
211 s1 = "abc def xyz"
212 print(len(s1))
213 print(s1.split())
214
215 s2 = "  abc abc  "
216 print(s2.strip())
217 print(s2.replace("abc","*"))
218 print(s2.find("abc"),s2.find("xxx"))
219 s3="Abc deF"
220
221 print(s3.startswith("Ab"))
222 print(s3.endswith("eF"))
223 print(s3.lower())
224 print(s3.upper())
225 s4=","
226 a=["aa","bb","cc"]
227 print(s4.join(a))
228
229 '''
230 10. 更复杂的格式化输出
231 当需要用到更复杂的格式化输出时，现查即可。可以参考：
232
233 更复杂的输出格式
234 printf 风格的字符串格式化
235
236 11. 作业题扩展内容
237 作业的评测器会自动忽略每一行的行末空格，所以行末输出多余空格也视为正确。
238 s.isdigit()：当字符串s不是空字符串，且包含的所有字符都是数字时返回True，否则返回False。
239 a, b = ["abc", "def"]这种写法可以将"abc"赋值给第一个变量a，将"def"赋值给第二个变量b。
240 s.rfind("abc")查询某个子串在s中最后一次出现的下标；如果不存在，则返回-1。
241 当不知道读入的具体行数时，可以采用如下方法一次性读取所有行：
242 from sys import stdin
243
244 for line in stdin.readlines():
245     print(line.strip()) # strip()是为了去掉行末的回车
246
247 '''
248

```

```
249 '''
250 在 Python 中，字符串的格式化输出有多种方法，包括传统的 `%` 格式化、`str.format()` 方法
    以及 f-string（格式化字符串字面值）。下面分别介绍这些方法及其用法：
251
252 ### 1. `%` 格式化
253 这是最早的一种字符串格式化方法，类似于 C 语言中的 `printf` 函数。
254
255 **语法：**
256
257 formatted_string = "格式字符串" % (变量1, 变量2, ...)
258
259
260 **示例：**
261
262 name = "Alice"
263 age = 30
264 message = "My name is %s and I am %d years old." % (name, age)
265 print(message) # 输出: My name is Alice and I am 30 years old.
266 '''
267
268 ### 2. `str.format()` 方法
269 这是一种更灵活的字符串格式化方法，支持更多的格式选项。
270
271 **语法：**
272
273 formatted_string = "格式字符串".format(变量1, 变量2, ...)
274 '''
275
276 **示例：**
277
278 name = "Alice"
279 age = 30
280 message = "My name is {} and I am {} years old.".format(name, age)
281 print(message) # 输出: My name is Alice and I am 30 years old.
282 '''
283
284 你也可以使用关键字参数来指定变量的位置：
285
286 message = "My name is {name} and I am {age} years old.".format(name="Alice",
    age=30)
287 print(message) # 输出: My name is Alice and I am 30 years old.
288 '''
289
290 ### 3. f-string（格式化字符串字面值）
291 这是 Python 3.6 引入的一种新的字符串格式化方法，使用起来非常简洁和直观。
292
293 **语法：**
```

```
294
295 formatted_string = f"格式字符串 {变量1} {变量2}"
296 ```
297
298 **示例: **
299
300 name = "Alice"
301 age = 30
302 message = f"My name is {name} and I am {age} years old."
303 print(message) # 输出: My name is Alice and I am 30 years old.
304 ```
305
306 f-string 还支持表达式和函数调用:
307
308 x = 10
309 y = 20
310 message = f"The sum of {x} and {y} is {x + y}."
311 print(message) # 输出: The sum of 10 and 20 is 30.
312 ```
313
314 ### 其他格式化选项
315 无论使用哪种方法, 都可以使用各种格式化选项来控制输出的格式。例如, 控制浮点数的小数位数、对
    齐方式等。
316
317 **示例: **
318 ```python
319 value = 123.456789
320 formatted_value = f"The value is {value:.2f}" # 保留两位小数
321 print(formatted_value) # 输出: The value is 123.46
322
323 number = 42
324 formatted_number = f"The number is {number:05d}" # 补零到5位
325 print(formatted_number) # 输出: The number is 00042
326
327 text = "hello"
328 formatted_text = f"The text is {text:>10}" # 右对齐, 总长度为10
329 print(formatted_text) # 输出: The text is      hello
330 ```
331
332 '''
333
334 #pycharm撤销操作 CTRL+Z
```

## 作业

```
1 #760. 字符串长度
2 a = str(input())
3 b = len(a)
4 print(b)
5
6 # 761. 字符串中的数字个数
7 a = str(input())
8 b = 0
9 for i in range(len(a)):
10     if "9">=a[i]>="0":
11         b+=1
12 print(b)
13
14 #765. 字符串加空格
15 a = str(input())
16 b = " "
17 print(b.join(a))
18
19 #769. 替换字符
20 a =str(input())
21 b =str(input())
22 print(a.replace(b,"#"))
23
24 #773. 字符串插入
25
26 from sys import stdin
27 for line in stdin.readlines():
28     a, b = line.strip().split()# 去除每行末尾的换行符并分割字符串
29     x = max(a)# 找到字符串 a 中字典序最大的字符
30     y = a.find(x)+1
31     c = a[:y]
32     d = a[y:]
33     print(c+b+d)
34
35 #772. 只出现一次的字符
36 a = str(input())
37
38 x = [0 for i in range(26)]
39
40 for c in a:
41     x[ord(c) - 97]+=1
42 for c in a:
43     if x[ord(c) -97] == 1:
44         print(c)
45         break
46 else:
47     print("no")
```

```

48
49 #764. 输出字符串
50 a = input()
51 c = list(a)
52 for i in range(len(a)):
53     c[i] = chr(ord(a[i]) + ord(a[(i+1)%len(a)]))
54 nea = ''.join(c)
55 print(nea)
56
57 #771. 字符串中最长的连续出现的字符
58
59 史:
60 a = int(input())
61 for i in range(a):
62     min1 = 0
63     tmp = 0
64     c = 0
65     b = input()
66     len0 = len(b)
67     for j in range(1, len0):
68         if b[j - 1] == b[j]:
69             tmp += 1
70         else:
71             tmp += 1
72             if tmp > min1:
73                 min1 = tmp
74                 c = ord(b[j - 1])
75             tmp = 0
76     if tmp > min1 and b[len0-1]==b[len0-2]:
77         min1 = tmp
78         c = ord(b[len0 - 1])
79         min1+=1
80     print("%s %d" % (chr(c), min1))
81
82 yxc:
83 n = int(input())
84
85 for i in range(n):
86     s = input()
87     max_c, max_t = ' ', 0
88     for j in range(len(s)): # 枚举起点
89         for k in range(j, len(s)): # 枚举终点
90             if s[k] != s[j]:
91                 break
92             cnt = k - j + 1
93             if cnt > max_t:
94                 max_c = s[j]

```

```
95         max_t = cnt
96     print(max_c, max_t)
97
98
99 #762. 字符串匹配
100 a = float(input())
101 b = input()
102 c = input()
103 x = 0
104 for i in range(len(b)):
105     if b[i]==c[i]:
106         x+=1
107 if x/len(b)>= a:
108     print("yes")
109 else:
110     print("no")
111
112 #768. 忽略大小写比较字符串大小
113 a = input().lower()
114 b = input().lower()
115 if a > b:
116     print(">")
117 elif a < b:
118     print("<")
119 else:
120     print("=")
121
122
123 # 763. 循环相克令
124 T = int(input())
125 for i in range(T):
126     a, b = input().split()
127     x, y = 0, 0 #设置权重
128     if a == "Bear":
129         x = 1
130     elif a == "Gun":
131         x = 2
132     if b == "Bear":
133         y = 1
134     elif b == "Gun":
135         y = 2
136
137     if x == y:
138         print("Tie")
139     elif x == (y + 1) % 3:
140         print("Player1")
141     else:
```

```
142         print("Player2")
143
144 #766. 去掉多余的空格
145 a = input()
146 b = []
147 len0 = len(a)
148 for i in range(len0):
149     if a[i]==' ' and a[i-1]!=' ':
150         pass
151     if a[i]!=' ' and a[i+1]!=' ':
152         b.append(a[i])
153     elif a[i]==' ' and a[i+1]==' ':
154         pass
155     else:
156         b.append(a[i])
157 ne=''.join(b)
158 print(ne)
159
160 #767. 信息加密
161 a = input()
162 b=[]
163 len0 = len(a)
164 for i in range(len0):
165     if 'a' <= a[i] <= 'z':
166         b.append(chr((ord(a[i])-ord('a')+1)%26+ord('a'))))
167     elif 'A' <= a[i] <= 'Z':
168         b.append(chr((ord(a[i]) - ord('A') + 1) % 26 + ord('A'))))
169     else:
170         b.append(a[i])
171 ne = ''.join(b)
172 print(ne)
173
174 #770. 单词替换
175
176 史:
177 a = input()
178 b = input()
179 c = input()
180 res = a.split(' ')
181 for i in range(len(res)):
182     if res[i] == b:
183         res = c
184 for i in res:
185     print(i,end=' ')
186
187 ysc:
188 s = input()
```



```
189 a = input()
190 b = input()
191
192 for word in s.split():
193     if word == a:
194         print(b, end=' ')
195     else:
196         print(word, end=' ')
197
198 #774. 最长单词
199
200
201 史:
202 a = input()
203 b = []
204 for i in range(len(a)-1):
205     b.append(a[i])
206 c = ''.join(b)
207 res = c.split(' ')
208 x = 0
209 tmp = 0
210 re = 0
211 for i in range(len(res)):
212     tmp = len(res[i])
213     if tmp > x:
214         x = tmp
215         re = i
216     tmp=0
217 print(res[re])
218
219 yxc:
220 s = input()[:-1] # 去掉末尾的句号
221
222 res = ""
223 for word in s.split():
224     if len(word) > len(res):
225         res = word
226
227 print(res)
228
229 #775. 倒排单词
230 a = input()
231 res = a.split(' ')
232 len0 = len(res)
233 for i in range(len0-1,-1,-1):
234     print(res[i],end=' ')
235
```

```
236
237 #776. 字符串移位包含问题
238 a, b = input().split()
239 if len(a) < len(b):
240     a, b = b, a
241
242 for i in range(len(a)):
243     a = a[1:] + a[0]
244     if a.find(b) != -1:
245         print("true")
246         break
247 else:
248     print("false")
```

## 六、函数

### 笔记

```
1 '''
2 在Python中，函数跟普通的变量是类似的，只不过它存储的不是值，而是一段逻辑。理解函数，最重要的是理解代码的执行顺序。——闫学灿
3
4 当代码中需要重复使用相同的逻辑时，可以使用函数来减少代码量。当代码较长时，可以使用函数来将逻辑分割成若干部分，使代码结构更容易让人看懂。
5
6 Python中函数的用法非常多，80%的用法不常用，20%的用法常用。大家不要把精力浪费在背完所有用法上，而要把主要精力放到最常用的20%的用法和代码逻辑上，至于另外80%不常用的用法，边用边查就行。
7
8 1. 函数基础
9 Python中一个典型的函数定义包括以下部分：关键字def、函数名称、由0个或多个形参组成的列表以及函数体。
10
11 1.1 编写函数
12 我们来编写一个求阶乘的函数。例如：
13
14 '''
15 def fact(n):
16     res = 1
17     for i in range(1,n+1):
18         res *= i
19     return res
20 '''
21 函数名称是fact，给它传入一个n，会返回n的阶乘。return语句负责结束函数并返回res的值。
```

```
22
23 1.2 调用函数
24 print("我们要计算5的阶乘，答案是：")
25 print(fact(5)) # 输出 120
26 print("计算结束啦！")
27 函数的调用完成两项工作：一是用实参初始化函数对应的形参，二是将控制权转移给被调用的函数。此
    时，代码原本的执行顺序被暂时中断，被调函数开始执行。等被调函数执行完后，再继续执行之前的
    代码。
28
29 1.3 形参和实参
30 实参指调用函数时传入的变量或常量，形参指定义函数时参数列表里的变量。
31
32 形参列表可以为空，例如：
33 '''
34 def f():
35     print("Hello World")
36 f()
37 '''
38 调用函数时会用实参去初始化形参，初始化的顺序有两种：
39
40 第一种是用位置实参来初始化形参。
41 顾名思义，实参会按位置关系来初始化形参，第一个实参初始化第一个形参，第二个实参初始化第二个
    形参，
42 依此类推。形参和实参的个数必须匹配。例如：
43 '''
44 f(1, True, "Python", 4.2) # 输出 a = 1, b = True, c = Python, d = 4.2
45 f(1, True, "Python", 4.2, 3) # 会报错，因为实参个数多于形参
46 f(1, True, "Python") # 会报错，因为实参个数少于形参
47
48 '''
49 第二种是用关键字实参来初始化形参。此时实参不再按位置关系来初始化形参，而是按变量名初始化。
    例如：
50 '''
51 # f()的定义如上所述
52 f(b=1, c=True, a="Python", d=4.2) # 输出 a = Python, b = 1, c = True, d = 4.2
53
54 #两种方式也可以混合使用，但是位置实参一定要放到关键字实参之前。例如：
55
56 # f()的定义如上所述
57 f(1, 2, d="Python", c=4.2) # 输出 a = 1, b = 2, c = 4.2, d = Python
58 f(1, b=3, "Python", d=4.2) # 会报错，因为位置实参位于关键字实参后面了。
59
60 '''
61 1.3.2 带默认值的形参
62 形参也可以设置默认值，
63 但所有带默认值的形参必须是最后几个。
64 当某些形参没有被初始化时，这些形参会使用默认值。例如：
```

```
65 '''
66 def f(a, b, c=3, d="Python"):
67     print("a =", a, end=" ")
68     print("b =", b, end=" ")
69     print("c =", c, end=" ")
70     print("d =", d)
71
72
73 f(1, 2) # c和d没有被初始化, 采用默认值。输出 a = 1, b = 2, c = 3, d = Python
74 f(1, b=2, d="AcWing") # c没有被初始化, 采用默认值。输出 a = 1, b = 2, c = 3, d =
    AcWing
75 '''
76 1.3.3 其它参数写法
77 其它参数写法用得不多, 想了解的同学可以参考函数定义详解。
78
79 1.4 变量的作用域
80 函数内定义的变量为局部变量, 只能在函数内部使用。
81
82 当需要修改用全局变量时, 需要用global关键字在函数内声明全局变量。例如: x = 1
83
84 '''
85 def f():
86     global x # 在函数内声明全局变量
87     x = 666
88     y = 777
89     print(x, y)
90 f() # 输出 666 777
91 print(x) # 会发现全局变量x也被修改了
92 print(y) # 会报错, 因为y是局部变量, 函数外无法使用
93 '''
94 1.5 嵌套定义函数
95 函数内部也可以定义函数。例如:
96 '''
97
98 def f():
99     def g(x): # 定义函数g()
100         x += 1
101         print(x)
102     g(5) # 调用函数g()
103
104
105 f() # 输出6
106 '''
107 1.6 pass语句
108 当函数定义完但还不想实现时, 可以用pass占位符, 来避免出现语法错误。例如:
109 '''
110 def f():
```

```
111     pass
112 '''
113 2. 参数传递
114 2.1 值传递
115 int、float、bool、字符串等采用值传递。
116
117 将实参的初始值拷贝给形参。此时，对形参的改动不会影响实参的初始值。例如：
118 '''
119 def f(y):
120     y = 5
121     print(y)
122 x = 10
123 f(x)
124 print(x) # 会发现x的值没变
125
126 '''
127 2.2 引用传递
128 列表采用引用传递。
129
130 将实参的引用传给形参，此时对形参的修改会影响实参的初始值。例如：
131 '''
132 def f(b):
133     for i in range(len(b)):
134         b[i] += 1
135
136
137 a = [0, 1, 2, 3, 4]
138 f(a)
139 print(a) # 会发现列表a中的每个数加了1
140 '''
141 3. return语句
142 return语句终止当前正在执行的函数并将控制权返回到调用该函数的地方，并返回结果。例如：
143 '''
144 def f(x):
145     if x == 1:
146         return # 不写返回值时，会返回None
147     if x == 2:
148         return 3 # 返回一个变量
149     if x == 3:
150         return 4, 5 # 返回多个变量
151
152
153 a = f(1)
154 b = f(2)
155 c, d = f(3)
156 e = f(4) # 没写return时，也会返回None
157 print(a, b, c, d, e) # 输出 None 3 4 5 None
```

```
158 '''
159 4. lambda表达式
160 lambda关键字可以创建匿名函数，目的是为了简化代码。
161 lambda 参数列表：表达式
162
163 可以对比下面两种写法，会发现lambda表达式的写法更短一些。
164 常与sort()函数配合使用，例如：
165 '''
166 pairs = [[1, "one"], [2, "two"], [3, "three"], [4, "four"]]
167
168 pairs.sort(key=lambda pair: pair[1]) # 每个元素使用第二个变量比较大小
169 print(pairs) # 输出: [[4, 'four'], [1, 'one'], [3, 'three'], [2, 'two']]
170 '''
171 5. 函数递归
172 在一个函数内部，也可以调用函数自身。这种写法被称为递归。
173
174 写递归函数可以从集合的角度来思考。理解递归函数的执行顺序可以用树的形式来思考。
175
176 例如，求解斐波那契数列第n项可以采用如下写法：
177 '''
178 def fib(n):
179     if n <= 2:
180         return 1
181     return fib(n - 1) + fib(n - 2)
182
183 print(fib(6)) # 输出 8
```

## 作业

```
1 #804. n的阶乘
2
3 def fact(n):
4     if n == 1:
5         return 1
6     else:
7         return n*fact(n-1)
8
9 n = int(input())
10 res = fact(n)
11 print(res)
12
13 #805. x和y的最大值
14 def max(x,y):
15     if x > y:
16         return x
```

```

17     else:
18         return y
19 x,y=map(int,input().split())
20 print(max(x,y))
21
22 # 808. 最大公约数
23 def gcd(a,b):
24     if a < b:
25         a,b=b,a
26     a = a - b
27     if a!=b:
28         return gcd(a,b)
29     else:
30         return a
31 a,b=map(int,input().split())
32 print(gcd(a,b))
33
34 #812. 打印数字
35 def print1D(a,size):
36     for i in range(size):
37         print(a[i],end=' ')
38 b,c=map(int,input().split())
39 a = list(map(int, input().split()))
40 print1D(a,c)
41
42 #816. 数组翻转
43 def reserve(a,b):
44     for i in range(b//2):
45         a[i],a[b-1-i]=a[b-1-i],a[i]
46 c,b=map(int,input().split())
47 a = list(map(int,input().split()))
48 reserve(a,b)
49 for i in range(c)
50     print(a[i],end=' ')
51
52 yxc:
53 def reverse(a, size):
54     i, j = 0, size - 1
55     while i < j:
56         a[i], a[j] = a[j], a[i]
57         i += 1
58         j -= 1
59
60
61 n, size = map(int, input().split())
62 a = list(map(int, input().split()))
63

```

```
64 reverse(a, size)
65
66 for x in a:
67     print(x, end=' ')
68
69
70 # 820. 递归求斐波那契数列
71 def f(n):
72     if n == 1:
73         return 1
74     elif n == 2:
75         return 1
76     else:
77         return f(n-1)+f(n-2)
78 n = int(input())
79 x = f(n)
80 print(x)
81
82 # 806. 两个数的和
83 def add(x, y):
84     return x + y
85
86
87 x, y = map(float, input().split())
88 print("%.2f" % add(x, y))
89
90 # 807. 区间求和
91
92 def sum(l, r):
93     res = 0
94     for i in range(l, r + 1):
95         res += i
96     return res
97
98
99 l, r = map(int, input().split())
100 print(sum(l, r))
101
102 # 809. 最小公倍数
103 def gcd(a,b):
104     if a < b:
105         a,b=b,a
106     a = a - b
107     if a!=b:
108         return gcd(a,b)
109     else:
110         return a
```



```

111 a,b=map(int,input().split())
112 print(a*b//gcd(a,b))
113
114 #两个数的乘积等于这两个数的最大公约数与最小公倍数的积。
115
116 yxc:
117 def lcm(a, b):
118     k = 1
119     while k % a != 0 or k % b != 0:
120         k += 1
121     return k
122
123
124 a, b = map(int, input().split())
125 print(lcm(a, b))
126

```

## 七、字典

### 笔记

```

1  '''
2  1. 元组
3  元组跟列表类似，只是不支持动态添加、删除元素，以及不能修改元素。
4
5  1.1 元组的初始化
6  元组需要用小括号括起来，中间的元素用逗号隔开。
7  注意，如果初始化只包含一个元素的元组，需要在该元素后添加逗号。
8
9  '''
10 a=()
11 b = (1,2)
12 c = 6,"python",3.14
13 '''
14 1.2 元组的解包
15 t = 12345, 54321, "Hello!" # 初始化一个元组
16 x, y, z = t # 将元组解包，将元组内的三个值按顺序赋值给x、y、z
17 print(x, y, z)
18 所以，2. 判断语句中学习的交换操作，本质上是元组的解包：
19
20 a, b = 3, 4 # 将元组(3, 4)解包，分别赋值给a、b
21 a, b = b, a # 将元组(b, a)解包，分别赋值给a、b
22 同样地，6. 函数中函数返回多个值，本质上也是返回了一个元组：
23 '''

```

```
24 def calc(x, y):
25     return x + y, x * y  # 等价于 return (x + y, x * y)
26
27
28 x, y = 3, 4
29 s, p = calc(x, y)  # 将(x + y, x * y)解包, 分别赋值给s、p
30 print(s, p)
31 '''
32 1.3 元组的比较运算
33 元组和列表均支持比较运算符: ==、!=、>、<、>=、<=等, 按字典序进行比较。
34
35 1.4 元组的其他操作
36 元组的下标访问元素、循环遍历、切片、加法和乘法运算等操作, 都与列表相同。
37
38 2. 集合
39 集合是Python中最常用的数据结构之一, 用来存储不同元素。
40 注意, 集合中的元素是无序的。
41
42 2.1 集合的初始化
43 创建集合用花括号或set()函数。注意: 创建空集合只能用set(), 不能用{}, 因为{}创建的是空字典, 会在下一小节里介绍字典。
44
45 集合常见的初始化方式:
46
47 '''
48 basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}  # 会自动去除重复元素
49 print(basket)  # 重复的元素已经去除了
50
51 s = set()  # 初始化一个空列表
52 print(s)
53
54 a = [1, 2, 1, 3, 1]
55 b = set(a)  # 将列表转化成集合, 一般是为了去重。
56 c = list(b)  # 将集合转化回列表
57 print(b, c)
58
59 x = "abracadabra"
60 a = set(x)  # 将字符串中的每个字符存到集合中
61 b = str(a)  # 注意, 这里并不能将集合转化回原字符串, 而是用格式化表示集合中的内容
62 print(a, b)
63 '''
64 2.2 集合的常用操作
65 假设a表示一个集合。
66
67 len(a) 返回集合中包含的元素数量。
68 a.add(x) 在集合中添加一个元素。
```

```
69 a.remove(x) 删除集合中的x，如果x不存在，则报异常。
70 a.discard(x) 删除集合中的x，如果x不存在，则不进行任何操作。
71 x in a 判断x是否在a中。
72 x not in a 判断x是否不在a中。
73 例如：
74 '''
75 a = {1, 2, 3}
76
77 print(len(a)) # 输出3
78
79 a.add(4)
80 print(a) # 输出 {1, 2, 3, 4}，注意集合中的元素是无序的。
81
82 a.remove(2)
83 print(a) # 输出 {1, 3, 4}
84
85 # a.remove(5) # 因为5不存在，所以会报异常
86 a.discard(5) # 因为5不存在，所以不进行任何操作
87 print(a) # {1, 3, 4}
88 '''
89 2.3 使用for循环遍历集合
90 类似于列表，集合也可以用for ... in ...的形式遍历。例如：
91 '''
92 a = {1, 2, 3}
93
94 for x in a: # 循环遍历整个集合
95     print(x, end=' ')
96
97 '''
98 3. 字典
99 字典是Python中最常用的数据结构之一，用来存储映射关系。
100 注意，字典中的元素是无序的。
101 不同于列表，字典是以key进行索引的，可以将每个key映射到某个value。
102 key可以是任何不可变类型，常用可以作为key的类型有数字和字符串。
103 列表因为是可变的，所以不能作为key。value可以是任意类型。
104
105 3.1 字典的初始化
106 创建字典用花括号或dict()函数。
107
108 '''
109 tel = {'jack': 4098, 'sape': 4139} # 创建一个字典
110 print(tel) # 输出 {'jack': 4098, 'sape': 4139}
111
112 a = dict() # 创建一个空字典
113 a[123] = "abc" # 在字典中插入一个key-value对
114 a[456] = "def" # 在字典中插入一个key-value对
115 print(a) # 输出 {123: 'abc', 456: 'def'}
```

```
116
117 b = list(a)  # 将字典的关键字转化成列表
118 print(b)  # 输出[123, 456]
119 '''
120 3.2 字典的常用操作
121 假设a表示一个字典。
122
123 len(a): 返回字典中的元素对数。
124 a[x]: 获取关键字x对应的值, 如果x不存在, 会报异常。
125 a.get(x): 获取关键字x对应的值, 如果x不存在, 会返回None, 不会报异常。
126 a.get(x, y): 获取关键字x对应的值, 如果x不存在, 会返回默认值y, 不会报异常。
127 a[x] = y: 在字典中插入一对元素, 如果关键字x已存在, 则将它之前映射的值覆盖掉。
128 del a[x]: 删除关键字x对应的元素对, 如果x不存在, 会报异常。
129 x in a: 检查字典中是否存在关键字x。
130 x not in a: 检查字典中是否不存在关键字x。
131 a.keys(): 返回字典的所有key。
132 a.values(): 返回字典的所有value。
133 a.items(): 返回字典的所有由key和value组成的元组。
134 例如:
135 '''
136 a = {'abc': 1, 'def': 2, 'python': 3}  # 初始化一个字典
137
138 print(len(a))  # 输出3
139 print(a['def'])  # 输出2
140 print(a.get('def'))  # 输出2
141 print(a.get('xyz', 5))  # 因为'xyz'不存在, 所以输出默认值5
142
143 a['hello'] = 4  # 插入一对元素 'hello' -> 4
144 print(a)  # 输出{'abc': 1, 'def': 2, 'python': 3, 'hello': 4}
145
146 a['def'] = 5  # 更新'def'映射的值
147 print(a['def'])  # 输出5
148
149 del a['python']  # 删除关键字'python'
150 print(a)  # 输出{'abc': 1, 'def': 5, 'hello': 4}
151
152 print('hello' in a)  # 输出True
153 print(a.keys())  # 输出dict_keys(['abc', 'def', 'hello'])
154 print(a.values())  # 输出dict_values([1, 5, 4])
155 print(a.items())  # 输出dict_items([('abc', 1), ('def', 5), ('hello', 4)])
156 '''
157 3.3 使用for循环遍历字典
158 类似于列表, 字典也可以用for ... in ...的形式遍历。例如:
159 '''
160 a = {'abc': 1, 'def': 2, 'python': 3}  # 初始化一个字典
161
162 for k in a:  # 遍历key
```

```

163     print(k, end=' ')
164 print() # 输出回车
165
166 for k in a.keys(): # 遍历key
167     print(k, end=' ')
168 print() # 输出回车
169
170 for v in a.values(): # 遍历value
171     print(v, end=' ')
172 print() # 输出回车
173
174 for k, v in a.items(): # 遍历key-value对
175     print("(%s, %d) " % (k, v), end=' ')
176 print() # 输出回车
177 '''
178 4. 作业题扩展内容
179 map()也可以用for ... in ...的形式遍历。
180 例如: for x in map(int, input().split())可以遍历一行内用空格隔开的每个整数。
181 map()函数的返回值也可以直接转化成set(),
182 例如: set(map(int, input().split()))可以将一行用空格隔开的整数存到set()中。
183 sorted()函数可以将列表、元组、集合、字典排序,并返回一个新列表。
184 如果对字典排序,则返回字典所有key排序后的列表。
185 '''
186

```

## 作业

```

1 # 4317. 不同正整数的个数
2 n = int(input())
3 s = set(map(int, input().split()))
4 s.discard(0)
5 print(len(s))
6
7 # 5302. 字符串赋值
8 n = int(input())
9 d={}
10 for i in range(n):
11     a,b=input().split()
12     b=int(b)
13     d[a]=b
14
15 m=int(input())
16 for i in range(m):
17     s = input()
18     print(d.get(s,-1))

```

```

19
20 # 3207. 门禁系统
21 n = int(input())
22 cnt = {}
23 for x in map(int, input().split()):
24     if x not in cnt:
25         cnt[x] = 0
26     cnt[x] += 1
27     print(cnt[x], end=' ')
28
29 # 3213. 数字排序
30 n = int(input())
31
32 cnt = {}
33 for x in map(int, input().split()):
34     if x not in cnt:
35         cnt[x] = 0
36     cnt[x] += 1
37
38 a = list(cnt.items())
39 a.sort(key=lambda item: (-item[1], item[0])) # 双关键字排序
40
41 for k, v in a:
42     print(k, v)

```

## 八、类...

```

1 '''
2 1. 类
3 类可以将变量、函数打包在一起，让代码在逻辑上更加清晰。
4 类名称一般采用驼峰命名法，函数一般采用下划线命名法。
5
6 类中函数的第一个参数都是self，用来调用类本身的变量和函数。
7 当调用类中函数的时候，第一个参数self不需要自己传递，Python会自动传递这个参数。
8
9 1.1 类的定义
10
11 '''
12 class Hero:
13     hero_count = 0 #在类中是共享的--类变量，用类变量时，习惯用类名来访问
14     def __init__(self, name, level=10): #构造函数
15         self.name = name #实例变量
16         self.level = level
17         print("Hero %s has been created." % name)

```

```

18         Hero.hero_count += 1
19
20     def __str__(self):
21         return "Hero: %s" %self.name
22
23     def greet(self):#问候
24         print("%s Hi"%self.name)
25
26     def move(self):#移动
27         print("%s:Move!"%self.name)
28     def get_level(self):#获取英雄的等级
29         return self.level
30     def next_level(self):
31         return self.get_level()+1
32
33 zeus = Hero("Zeus")
34 athena = Hero("Athena",6)
35
36 zeus.greet()#自动传入宙斯
37 athena.greet()
38 print(zeus.name,athena.name)
39 print(zeus.hero_count,athena.hero_count)
40 athena.move()
41 print(str(zeus))
42 '''
43

```

## 1.2 类变量和实例变量

每个类可以创建任意多实例。例如上面的Hero类，可以创建zeus和athena等实例。

类变量由所有实例共享，一般通过类名访问，例如Hero.hero\_count。

实例变量与每个具体的实例绑定，一般通过具体实例来访问，例如zeus.name。

## 1.3 类的继承

子类可以继承父类的变量和函数。

self可以调用自身和父类中的变量和函数，super()可以调用父类中的函数。

如果子类 and 父类的变量或函数重名，优先使用子类的变量和函数。

```

56 '''
57 class Hero:
58     hero_count = 0 # 类变量
59
60     def __init__(self, name, level=10): # 构造函数
61         self.name = name
62         self.level = level
63         print("Hero %s has been created." % name)
64         Hero.hero_count += 1

```

```

65
66     def __str__(self): # 定义str()函数的效果
67         return "Hero: %s" % self.name
68
69     def greet(self): # 问候
70         print("%s: Hi!" % self.name)
71
72     def move(self): # 移动
73         print("%s: Move!" % self.name)
74
75     def get_level(self): # 获取这个英雄的等级
76         return self.level
77
78     def next_level(self):
79         return self.get_level() + 1 # 调用类中的其他函数
80
81 class zeus(Hero): # 基类、父类
82     hero_name = 'zeus'
83
84     def __init__(self, level):
85         super().__init__(zeus.hero_name, level) # 调用父类函数
86
87     def greet(self):
88         print("%s: Hi(from child class)"%self.name)
89
90
91 class athena(Hero):
92     hero_name = 'athena'
93     def __init__(self, level):
94         super().__init__(athena.hero_name, level)
95
96     def greet(self):
97         print("%s: Hi(from child class)"%self.name)
98
99
100 Zeus = zeus(6)
101 Athena = athena(8)
102 print(Zeus.name, Athena.level, Hero.hero_count)
103 print(str(Zeus), str(Athena))
104 Zeus.greet()
105 Athena.greet()
106
107 '''
108 2. 异常处理
109 当某段代码出现异常时，代码会被终止。此时如果不想让代码终止，可以用try ... except ... 语句来处理异常。
110

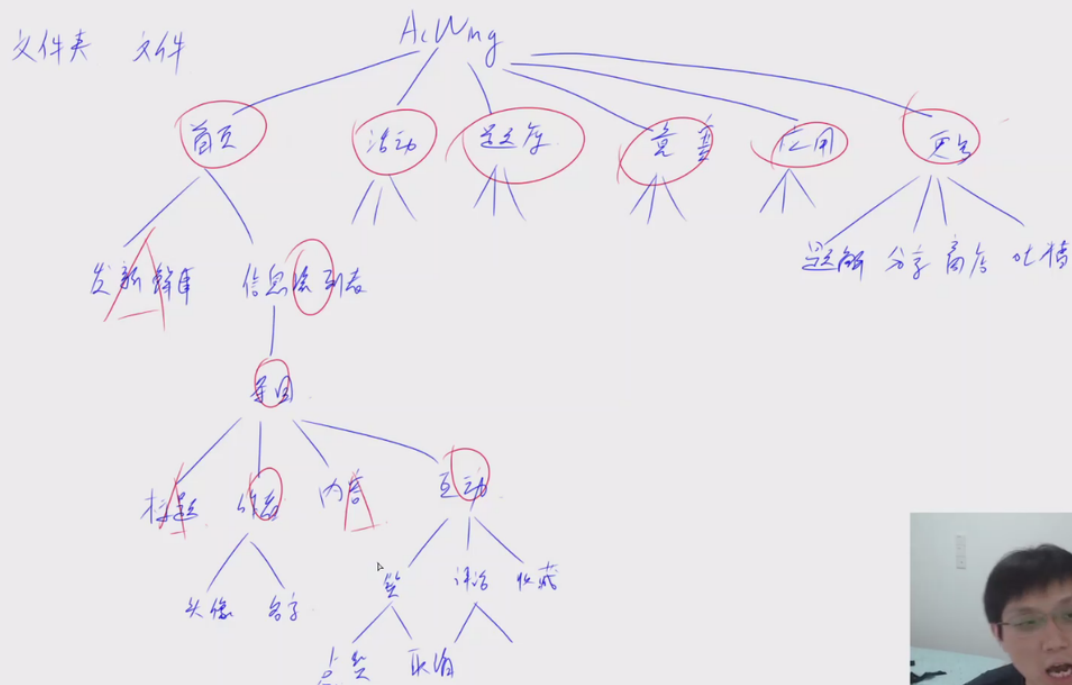
```



```
111 例如，将字符串转化成整数时，可能会出现异常：
112
113 '''
114 s = input()
115 try:
116     x = int (s)
117     print(x)
118 except Exception as e:
119     print(e)
120
121
122 x, y = map(int, input().split())
123
124 try:
125     z = x / y
126     print(z)
127 except Exception as e:
128     from traceback import print_exc
129     print(print_exc()) #可以输出错误的详细信息
130
131 print("Finished!")
132
133
134
```

PaintTool SAI Ver. 2 (64bit) 2022.09.10 - 新建画布1 (\*)

文件(F) 编辑(E) 图像(I) 图层(L) 选择(S) 尺寸(R) 滤镜(T) 视图(V) 窗口(W) 其他(O)

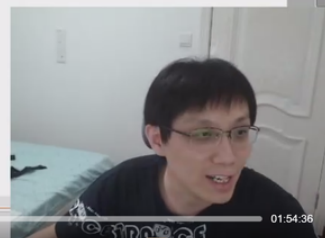
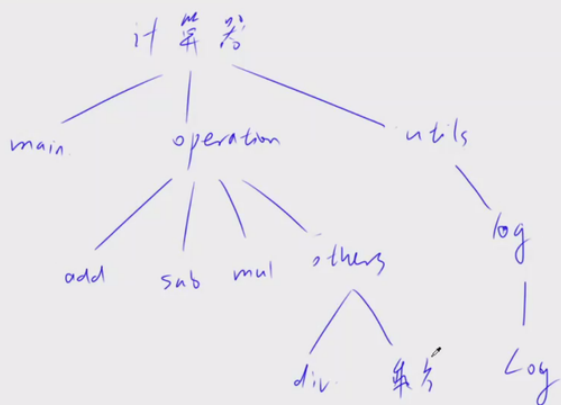


## 8. 类、异常处理、模块、包的管理、虚拟环境 - AcWing.mp4

13:44

PaintTool SAI Ver. 2 (64bit) 2022.09.10 - 新建画布1 (\*)

文件(F) 编辑(E) 图像(I) 图层(L) 选择(S) 尺寸(R) 滤镜(T) 视图(V) 窗口(W) 其他(O)



01:48:22

01:54:36

```
1 '''
2 3. 模块
3 当项目的逻辑越来越复杂时，把所有代码写到一个文件中会相当不方便。
4
5 此时就可以将不同代码放到不同的模块中。所有模块通过文件夹和文件组织成树的形式。
6
7 自定义模块的每个文件夹中需要包含一个空__init__.py文件，用来让Python识别出这是一个模块。
8
9 示例：
10
11 文件结构如上所示：
12 '''
13
14 utils/__init__.py 是空文件。
15
16 utils/log.py:
17
18 from datetime import datetime
19
20
21 class Log:
22     def __init__(self, text):
23         self.text = text
24
25     def info(self):
26         print(datetime.now().strftime("%Y-%m-%d %H:%M:%S"), "INFO:", self.text)
27 operation/__init__.py 是空文件。
28
29 operation/add.py:
30
31 from utils.log import Log
32
33
34 def add(a, b):
35     log = Log("Operation add")
36     log.info()
37     return a + b
38 operation/sub.py
39
40 import utils.log
41
42
43 def sub(a, b):
44     log = utils.log.Log("Operation sub")
45     log.info()
46     return a - b
```

```

47 operation/mul.py:
48
49 from utils.log import Log as MyLog
50
51
52 def mul(a, b):
53     log = MyLog("Operation mul")
54     log.info()
55     return a * b
56 operation/others.py:
57
58 from utils.log import Log
59
60
61 def div(a, b):
62     log = Log("Operation div")
63     log.info()
64     return a / b
65
66
67 def power(a, b):
68     log = Log("Operation power")
69     log.info()
70     return a ** b
71 main.py:
72
73 from operation.add import add
74 from operation.sub import sub
75 from operation.mul import mul
76 from operation.others import div, power
77
78 op = input()
79 a, b = map(int, input().split())
80
81 if op == '+':
82     print(add(a, b))
83 elif op == '-':
84     print(sub(a, b))
85 elif op == '*':
86     print(mul(a, b))
87 elif op == '/':
88     print(div(a, b))
89 elif op == '**':
90     print(power(a, b))

```

#### 4. 包的管理

写项目时，不会所有代码都自己实现，会用到很多网络上现成的包（模块）。

#### 4.1 通过命令行管理

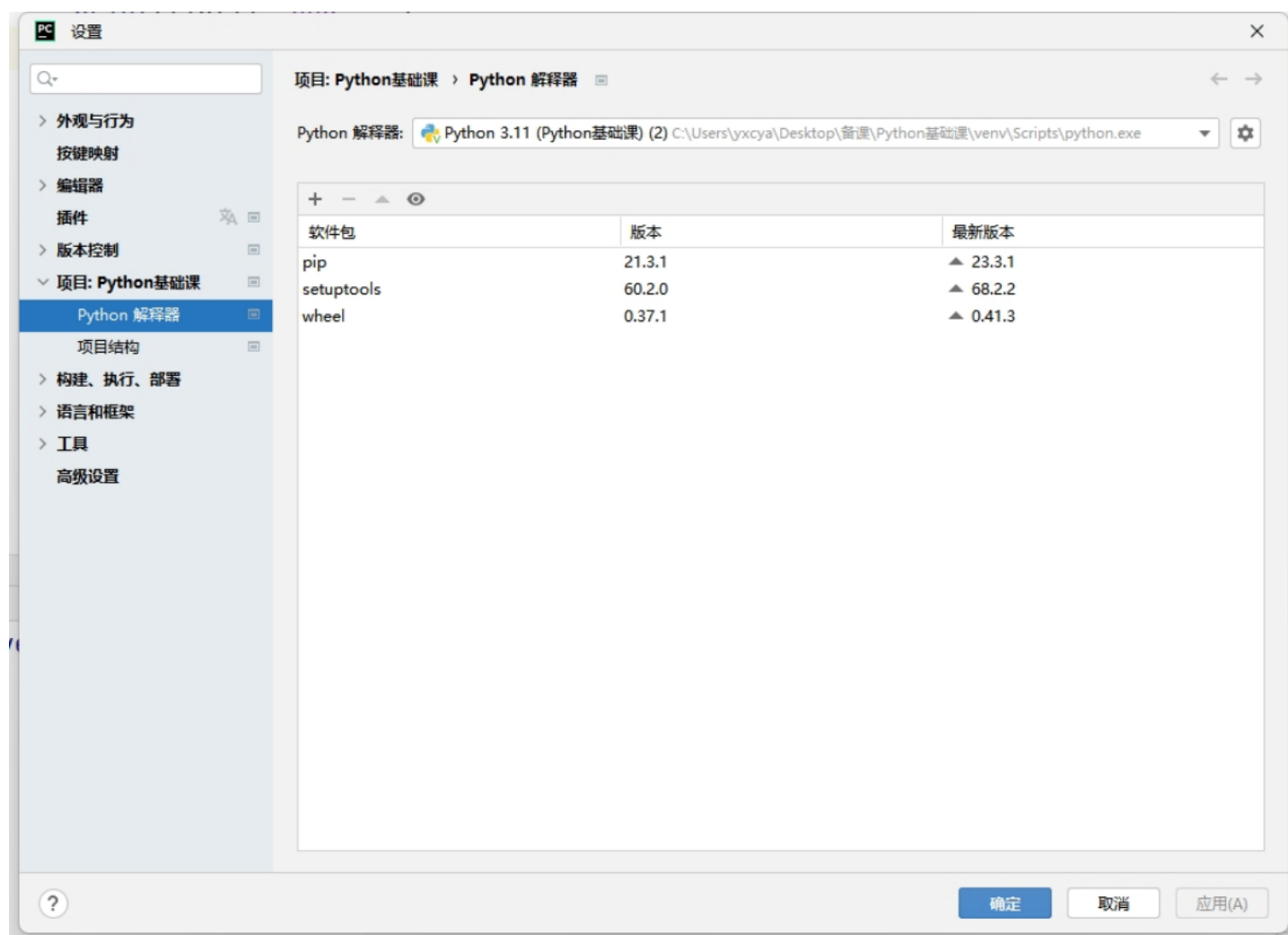
以安装pygame这个包为例：

安装包：pip3 install pygame --user

升级包：pip3 install pygame --user --upgrade

删除包：pip3 uninstall pygame

#### 4.2 通过IDE的图形化界面管理



#### 4.3 示例

下一章我们会用pygame包来开发迷宫项目。

现在我们可以先尝试一下这个包，用它来创建一个可关闭的窗口，名称是“Maze”：

```
import pygame

pygame.init()

screen = pygame.display.set_mode((700, 500)) # 窗口大小 700 * 500

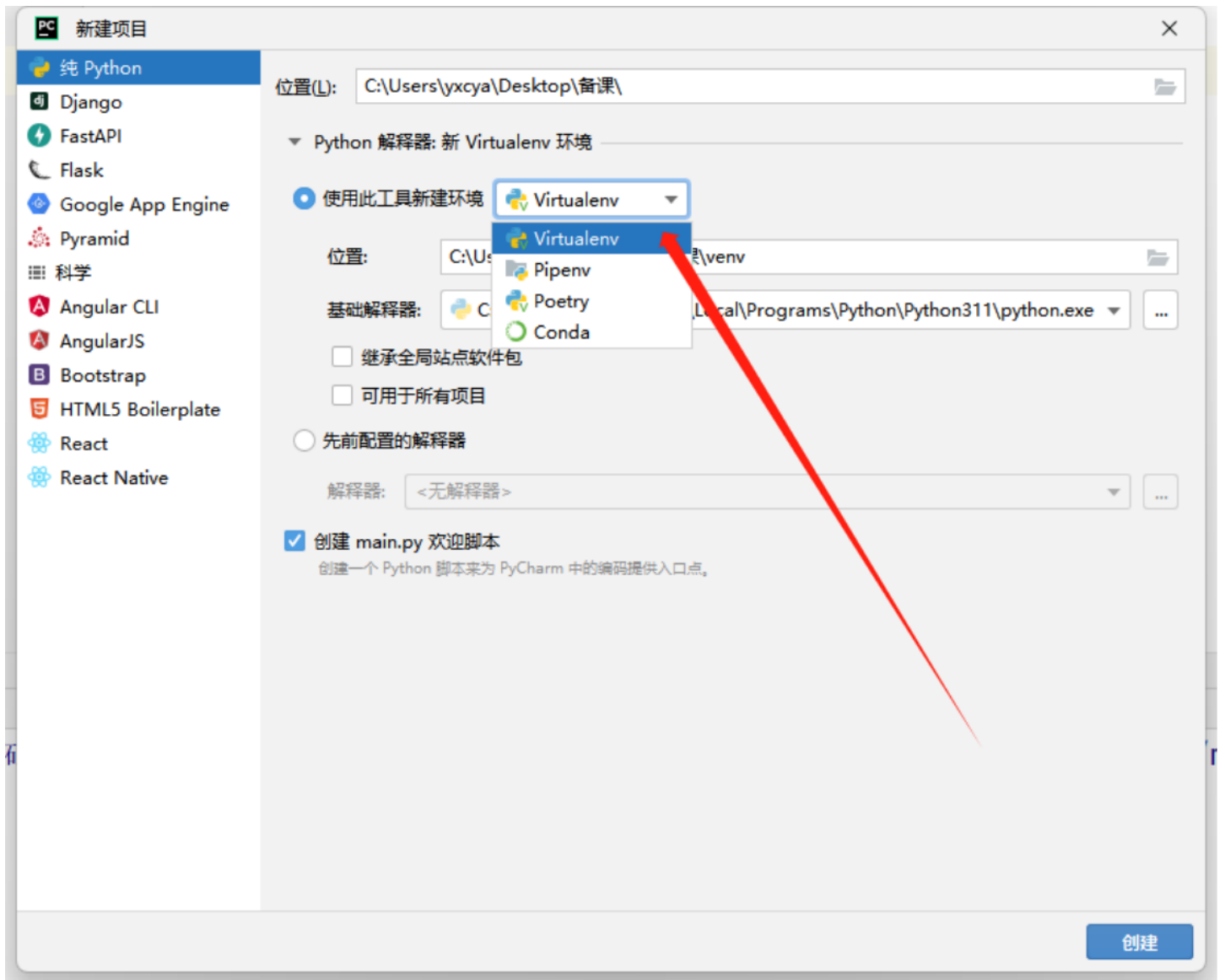
pygame.display.set_caption("Maze") # 窗口标题是 Maze

done = False
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: # 如果检测到窗口被关闭了，就终止程序
            done = True
```

## 5. 虚拟环境

每个虚拟环境主要用来存放一系列不同版本的包。

在windows或mac下开发时，一台电脑上可能会有多个Python项目，不同项目可能会依赖不同版本的包，为了不让不同项目之间产生冲突，可以将每个项目放到单独的虚拟环境中。



虚拟环境主要用于本地开发，在云端Linux服务器上部署项目时，一般采用docker来管理环境，不需要使用虚拟环境env。