



# P231 课时09 简单算法及其程序实现

1.C 2.  $m//3$  3. B 4. (1) 枚举算法 (2)  $i\%623==0$  5.D

6. ①  $0 < x < 30$  ②  $y = \text{math.sin}(x)$

7. (1) A (2) ①  $le/4$  ②  $(le-2h)/2$  或  $(le+d**0.5)/4$

8. (1) 9 (2)  $\text{sum}-=\text{down}$

9. (1) ① `return True` ②  $n=n+\text{dic}[\text{ch}]$  ③  $\text{flag}=\text{isprime}(s)$  (2) 这是一个素单词 (3) B

10. (1) B (2) `int(x[6:])` 或 `int(x[-2:])` (3)  $\text{tmp}=\text{tosec}(t[i+1])-\text{tosec}(t[i])+1$  (4)  $c[i]==\text{cmax}$

11. (1) 8 (2) ①  $\text{ord}(i)-97$  或  $\text{ord}(i)-\text{ord}('a')$  ② `width[index]` (3) `s=input("s: ")`

12. (1) 1 (2) ①  $m=\text{light}[i-1]+\text{light}[i-2]$  ②  $\text{count}/n > 0.5$  或  $\text{count} > n//2$  或  $\text{count} > n/2$  ③  $\text{light}[j]=1-\text{light}[j]$

13. (1) ①  $s=s+i$  ②  $k+1$  ③ `data[p:n-(k-p)]` (2) B

14. (1) 相同 (2) ①  $s=s+i$  ②  $i=i+1$  或  $i+=1$  ③  $c1=="?"$  or  $c2=="?"$

15. (1) A (2) ①  $m=n**2$  或  $m=n*n$  ②  $i*=10$  或  $i=i*10$  ③  $a[i]+a[i+1]==n$

16. (1) ①  $d[i]*d\_num[i]$  ②  $\text{money} < \text{sum}$  ③  $n=\text{int}(\text{sum}/d[i])$  或  $n=\text{sum}//d[i]$  (2) `range(len(d)-1,-1,-1)` (3) 8

# 一、解析算法

1. 计算球表面积的过程如下：

①输入球半径 $r$ ；

② $\pi = 3.14$ ；

③计算球表面积为 $4\pi r^2$ ；

④输出球表面积 $S$ 。

求解该问题适合使用的算法是 ( C )

A. 排序算法

B. 枚举算法

C. 解析算法

D. 查找算法

解析算法的基本思想是根据问题的前提条件与所求结果之间的关系，找出求解问题的数学表达式，并通过对其计算来实现问题的求解。

2. 有一对兔子，从出生后第3个月起每3个月生一对兔子，小兔子长到第3个月后每3个月又生一对兔子。假如兔子都不会死亡，求第m个月的兔子总数。实现上述功能的Python程序如下，请在划线处填入合适的代码。

```
t=2
m=int(input("输入第几个月："))
n=_____1
tz=2**(n+1)
#f格式化输出，将{}中的变量值代入输出
print(f"第{m}个月兔子的总数为{tz}")
```

使用解析算法，首先分析已知条件和所求结果之间的关系。根据题目给出的条件，列出下表：

月份	1	2	3	4	5	6	7	8	9	...
兔子数量	2	2	4	4	4	8	8	8	16	...

可以看到每当月份为3的倍数时，兔子的数量翻倍，所以可以列出关系式：兔子数量=2\*\*(月份//3+1)，故n=m//3。

## 二、枚举算法

3. 下列问题中，适合使用枚举算法解决的有 ( **B** )

- ①找出50内的所有素数
- ②计算已知半径的圆的周长
- ③列举10000以内所有13的倍数
- ④某次测试成绩排名
- ⑤在某字符串中找出所有数字

A. ①②③

B. ①③⑤

C. ②③④

D. ②④⑤

枚举算法的基本思想是把问题所有可能的解一一列举，然后判断每一个列举出的可能解是否为正确的解。本题中，只有①③⑤适合使用枚举算法解决，②适合使用解析算法，④适合使用排序算法。

4. 小明设置了一个5位数字的密码，可现在他只记得4\*\*9\*，“\*”位置的数字被遗忘。已知该数字密码是623的倍数，设计Python程序帮助小明找出所有可能的密码。

```
li=[]  
for i in range(40090, 50000):  
    if str(i)[0]=="4" and str(i)[3]=="9" and 枚举算法:  
        #append()用于在列表末尾添加元素  
        li.append(i)  
print(li)
```

请回答下列问题：

- (1) 该程序使用的算法是  $i\%623==0$ 。
- (2) 请在划线处填入合适的代码。

(1) for循环遍历范围内的所有可能解，if语句进行判断，使用的是枚举算法。 (2) 由题意可知，该5位数4\*\*9\*是623的倍数。

5. 著名的鸡兔同笼问题“今有雉兔同笼，上有三十五头，下有九十四足，问雉兔各几何？”，解决该问题的Python程序如下：

```
for m in range(0, 48):
```

```
    for n in range(0, 24):
```

```
        if m+n==35 and 2*m+4*n==94:
```

```
            print("鸡有"+str(m)+"只，兔子有"+str(n)+"只")
```

根据描述，一只兔子4只脚，鸡和兔子一共94只脚，因此兔子的数量n不会超过23，若将n的范围修改为range(0, 36)，会增加n的枚举数量，不能优化算法。

下列关于该程序的说法，**不正确**的是 ( **D** )

A. 该程序采用了枚举算法的思想

B. 将第一行代码改为“for m in range(0, 36)”可以减少枚举次数，优化程序

C. 执行该程序段后，输出“鸡有23只，兔子有12只”

**✓** D. 为了优化程序，可将m和n的范围都修改为range(0, 36)

6. 有如下分段函数:

$$y = \begin{cases} \cos(x) & (0 < x < 30) \\ x & (30 \leq x < 50) \\ \sin(x) & (x \geq 50) \end{cases}$$

根据分段函数设计如下Python程序, 请在划线处填入合适的代码。

```
import math
x=int(input("输入一个大于0的数x: "))
if _____ ① 0<x<30 或 x>0 and x<30
    y=math.cos(x)
elif 30<=x<50:
    y=x
else:
    _____ ② y=math.sin(x)
print("运算结果为: ", y)
```

根据分段函数, 将x的范围、y的表达式用Python语句表达即可。

7. 现使用一根长度为 $le$ 厘米的铁丝，制作一个面积为 $s$ 平方厘米的矩形框，计算该矩形框应有的高 $h$ 和宽 $w$ 。已知 $h$ 、 $w$ 、 $le$ 、和 $s$ 之间的关系： $2 \times h + 2 \times w = le$ ， $h \times w = s$ 。实现上述功能的Python程序如下，程序运行界面如图所示。

请输入铁丝的长度：4

请输入矩形框面积：1

长度为4厘米的铁丝可以制作面积为1平方厘米的正方形框，高为1.0厘米宽为1.0厘米

请输入铁丝的长度：12

请输入矩形框面积：8

长度为12厘米的铁丝可以制作面积为8平方厘米的矩形框，高为2.0厘米，宽为4.0厘米

```
le=int(input("请输入铁丝的长度："))
s=int(input("请输入矩形框面积："))
d=le*le-16*s
if d<0:
print(f"无整数解，长度为{le}厘米的铁丝制作不了面积为{s}平方厘米的矩形框")
elif d==0:
h=w=                  ① le/4                  
print(f"长度为{le}厘米的铁丝可以制作面积为{s}平方厘米的正方形框，高为{h}厘米，宽为{w}厘米")
else:
h=(le-d**0.5)/4
w=                  ② (le+d**0.5)/4                  
print(f"长度为{le}厘米的铁丝可以制作面积为{s}平方厘米的矩形框，高为{h}厘米，宽为{w}厘米")
```

请回答下列问题：

(1) 该程序段采用的算法是           A           (单选，填字母：A. 解析算法 / B. 枚举算法)。

(2) 请在划线处填入合适的代码。

(1) 解析算法的基本思想是指根据问题的前提条件与所求结果之间的关系，找出求解问题的数学表达式，并通过表达式的计算来实现问题的求解。在该程序段中，根据输入的整数  $le$  和  $s$ ，通过一系列的逻辑判断和数学运算，最终得到结果。 (2)

①根据输出判断条件和输出结果可知，该处为判断是否可以制作正方形框， $h$ 和 $w$ 的值为总长的四分之一，故填入代码为  $le/4$ 。②根据“ $2 \times h + 2 \times w = le$ ”可知， $w = (le - 2 \times h) / 2$ ，根据代码“ $h = (le - d ** 0.5) / 4$ ”代入可知， $w = (le + d ** 0.5) / 4$ 。

8. 有一口深度为10米的水井，井底有一只青蛙，它每天白天能够沿井壁向上爬2米，夜里则顺井壁向下滑1米，若青蛙从某个早晨开始向外爬，计算青蛙多少天能够爬出井口？编写如下Python程序解决这个问题，请回答下列问题：

```
high, up, down=10, 2, 1
```

```
sum, n=0, 0
```

```
while sum<high:
```

```
    n+=1
```

```
    sum+=up
```

```
    if sum<high:
```

```
        sum-=down 或 sum=sum-down
```

```
    else:
```

```
        print(n)
```

(1) 第一天，青蛙上爬2米，下滑1米，距离井口9米；第二天，青蛙上爬2米，下滑1米，距离井口8米；以此类推，第九天，青蛙上爬2米，跳出井口，故输出的结果是9。 (2) 划线处代码表示青蛙还未跳出井口时要执行的语句，应填入的代码为跳的总米数减去下滑的米数，即 $sum-=down$ 或 $sum=sum-down$ 。

(1) 执行该程序段后，输出的结果为 9。

(2) 请在划线处填入合适的代码。

9. [2025绍兴联考] 寻找素单词。素数是指大于1的自然数中，除了1和它本身以外不再有其他因数的自然数。例如2、3、5、7是素数。假如每个单词由小写字母组成，每个字母对应一个特定的值，字母a对应1，字母b对应2……以此类推，字母z对应26。若这个单词的字母值的总和是素数，则这个单词是素单词。编写Python程序，判定输入的单词是否为素单词。请回答下列问题：

(1) 实现上述功能的部分Python程序如下，请在划线处填入合适的代码。

```
def isprime(n): #判断n是不是素数
    for i in range( 2,int(n**0.5)+1 ):
        if n%i==0:
            return False
    return True
```

```
def strsum(word): #统计单词的字母值总和
```

```
dic={"a":1,"b":2,"c":3,"d":4,"e":5,"f":6,"g":7,"h":8,
     "i":9,"j":10,"k":11,"l":12,"m":13,"n":14,"o":15,"p":16,
     "q":17,"r":18,"s":19,"t":20,"u":21,"v":22,"w":23,"x":24,
     "y":25,"z":26}
```

```
n=0
for ch in word:
    ② n=n+dic[ch]
return n
word=input("请输入一个单词:")
s=strsum(word)
③ flag=isprime(s)
```

```
if flag:
    print("这是一个素单词")
else:
    print("这不是一个素单词")
```

(2) 若输入的单词是“run”，则输出的结果是“这是一个素单词”。

(3) 下列选项中，不可以填入加框处的是

B (单选，填字母)。

- A. 2, n
- B. 2, n+1
- C. 2, int(n\*\*0.5)+1
- D. 2, n//2+1

(1) ①在 `isprime` 函数中，当 `for` 循环结束都没有找到能整除 `n` 的数时，说明 `n` 是素数，应返回 `True`，故填入代码为 `return True`。②在 `strsum` 函数中，遍历单词中的每个字母，根据字典 `dic` 找到字母对应的数值并累加到 `n` 中，故填入代码为 `n=n+dic[ch]`。③处调用 `isprime` 函数判断字母值总和 `s` 否为素数，并将结果存储在 `flag` 中，故填入代码为 `flag=isprime(s)`。(2) 首先，根据字母与数值的对应关系，计算单词“run”的字母值的总和，字母“r”对应18，字母“u”对应21，字母“n”对应14，总和为53。因为53除了1和它本身以外不再有其他因数，是素数，故输出的结果是“这是一个素单词”。(3) `range(2, n+1)` 会让循环多执行一次，检查 `n` 本身，但 `n` 本身肯定能被自己整除，这不符合素数判断的逻辑，所以不可以填入加框处。

10. [2024.7浙江] 某 App 依据手机每秒钟屏幕的亮、熄，记录当天 00:00:00 至 23:59:59 之间手机持续亮屏的各时间段，并生成手机亮屏日志文件。若用 0 表示熄屏，1 表示亮屏，某天前 10 秒的屏幕状态为“0011011110”，则日志中记为“00:00:02, 00:00:03, 00:00:05, 00:00:08”，表示首个持续亮屏时间段的开始与结束时间点分别为“00:00:02”和“00:00:03”，时长为 2 秒，而第 2 个持续亮屏时间段时长为 4 秒。

编写 Python 程序读取某天的日志文件，计算并输出手机当天亮屏的总时长、单次持续亮屏时长的最大值和相应的时间段（若相应时间段有多个，则按时间顺序全部输出）。

(1) 若某天日志文件记录的前 6 个时间点如图所示，可知当天 10:00:00 该手机处于     **B**    （单选，填字母：A. 亮屏状态 / **B.** 熄屏状态）。

08:00:00,08:01:20,09:00:00,09:59:59,10:05:03,11:05:02

(2) 定义函数tosec(x)，参数x是形如“08:00:00”的字符串，表示时间点。该函数返回时间点x是当天的第几秒（约定00:00:00为当天第0秒）。请在划线处填入合适的代码。

```
def tosec(x):  
    '''从时间点x中提取时、分，转为整型后分别存储在变量h、m中，代码略'''  
    s= int(x[6:]) 或 int(x[-2:])  
    n=h*3600+m*60+s  
    return n
```

(3) 定义函数getdata(t)，该函数功能是依次计算每个持续亮屏时间段的时长并存入列表c，对应时间段存入列表d，返回c和d。请在划线处填入合适的代码。

```
def getdata(t):  
    i=0  
    c=[];d=[]  
    while i<len(t):  
        tmp=tosec(t[i+1])-tosec(t[i])+1  
        #在列表c末尾添加一个元素  
        c.append(tmp)  
        d.append(t[i]+"—" +t[i+1])  
        i+=2  
    return c,d
```

08:00:00,08:01:20,09:00:00,09:59:59,10:05:03,11:05:02

(4) 实现上述功能的主程序如下，请在划线处填入合适的代码。

''' 读取日志文件中各时间点，依次存入列表t，t形如["08:00:00",.....]，代码略'''

```
c, d=getdata(t)
tot=0;cmax=0
res=""
for i in range(len(d)):
    tot+=c[i]
    if c[i]>cmax:
        cmax=c[i]
        res=d[i]
    elif c[i]==cmax:
        res=res+", "+d[i]
    ,,,
```

(1) 每2个时间点为一组，分别表示亮屏的开始和结束时间，第2组中亮屏的结束时间为09:59:59，第3组的开始时间为10:05:03，因此10点为熄屏时间。  
 (2) 函数tosec的功能是将时间转换为秒，变量h和m分别表示时和分，因此变量s表示秒，将索引第6、7位上的数字串转换为整型。  
 (3) 函数getdata(t)是依次计算每个持续亮屏时间段的时长和持续时间，i表示t数组中索引，每2个时间点为一个持续的时间段，调用tosec函数将前后两个时间点(t[i+1]和t[i]分别表示结束和开始时间点)转换成对应的秒数，将两个秒数的差值加1，结果为持续时长tmp。  
 (4) cmax表示最长持续时间，c[i]表示当前的持续时长，若相应时间段有多个，需将c[i]连接到res中，按时间顺序全部输出。

输出当天亮屏的总时长tot

输出单次持续亮屏时长的最大值cmax和相应的时间段res

代码略

'''



(2) 实现上述功能的 Python 程序如下, 请在划线处填入合适的代码。

```
width=[4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6]
```

```
s=int(input("s: ")) s=input("s: ")
```

```
L=0
```

```
count=1
```

```
for i in s:
```

```
    index=_____ ① ord(i)-97
```

```
    L=L+width[index]
```

```
    if L>20:
```

```
        L=_____ ② width[index]
```

```
        count+=1
```

```
print("小写字母宽度列表 width 为:")
```

```
print(width)
```

```
print(f"写入字符串 s 共需 {count} 行, 最后一行的宽度为: {L}")
```

(3) 程序中加框处代码有错, 请改正。

(1) "b"对应的宽度为6, "a"对应的宽度为4, 由于第一个 "a"的宽度加上前面3个"b"的宽度超过20, 故"bbb"写入第一行, 宽度为6+6+6=18, "aa"写入第二行, 宽度为4+4=8。 (2) ①根据下一行代码可知, 变量index用于定位每个小写字母在列表width中的位置, 故填入代码为ord(i)-97。②处利用for循环对宽度L进行累加, 当加入某个字母的宽度后, 总宽度L超过20, 则将该字母的宽度重新赋值给总宽度L并开始下一行宽度的累加, 故填入代码为width[index]。 (3) 由后面的语句可知, s为字符串, 不需要用到int()函数, 故正确代码为s=input("s: ")。

12. [2025新阵地联盟] 某照明设备可控制 $n$ 盏 ( $n > 2$ ) 霓虹灯状态, 每盏灯的编号依次为 $0 \sim n-1$ , 状态数据存储在列表`light` (0表示灯灭, 1表示灯亮) 中。编写Python程序, 每隔固定时间控制霓虹灯状态并输出。

亮灭规则如下: 规则一, 某时刻从第3盏灯开始, 灯的状态取决于它的前两盏灯的状态之和 (若和为奇数则亮, 否则灭); 规则二, 若亮灯数量大于霓虹灯总数的50%, 则将每一盏灯的状态取反, 否则将第一盏灯状态取反。

重复上述规则, 请回答下列问题:

(1) 若`light`初始为`[1, 0, 0, 0, 0]`, 则经过规则一后, 编号为3的霓虹灯的状态值为 1 (选填: 0 / 1)。

(2) 实现上述功能的部分Python程序如下,请在划线处填入合适的代码。

```
#读取霓虹灯的总数,存入n,代码略
light=[0]*n #初始化霓虹灯的状态
while True:
    for i in range(2,n):
        ① m=light[i-1]+light[i-2]
        if m%2==0:
            light[i]=0
        else:
            light[i]=1
    #输出霓虹灯的当前状态,代码略
    #延时1秒,代码略
    count=sum(light)
    if ② count/n>0.5 或 count>n//2 或 count>n/2 或其他等价答案:
        j=0
        while j<n:
            ③ light[j]=1-light[j] 或 light[j]=abs(light[j]-1) 或其他等价答案
            j+=1
    #输出霓虹灯的当前状态,代码略
else:
    light[0]=1-light[0]
    #延时1秒,代码略
```

(1) 根据规则一,从第3盏灯(编号为2)开始,依次的状态值为1、1、0,故编号为3的霓虹灯的状态值为1。(2) ①变量m是前两盏灯的状态值之和,故填入代码为  $m=light[i-1]+light[i-2]$ 。②count是亮灯的总数,根据规则二和下面的代码可知,该处为亮灯数量大于霓虹灯总数的50%的情况,故填入代码为  $count/n>0.5$ 。③处将每一盏灯的状态取反,故填入代码为  $light[j]=1-light[j]$ 。

13. [2025杭州联考] 有一个长度为 $n$ 的列表`data`，每次操作只能选择删除列表的第一个元素或者最后一个元素，对这个列表进行 $k$ 次操作后，要使剩下的列表元素之和达到最大值。编写Python程序，程序运行界面如图所示，请回答下列问题：

请输入列表元素的个数：6

请输入删除的次数：2

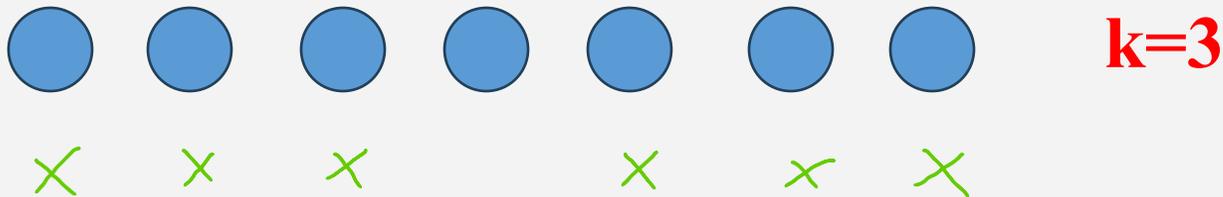
列表为：[15,20,2,15,18,7]

删除后，列表为：[20,2,15,18] 最大值为：55

(1) 实现上述功能的Python程序如下, 请在划线处填入合适的代码。

```
import random
def add(d):
    s=0
    for i in d:
        ① s=s+i
    return s
n=int(input("请输入列表元素的个数: "))
k=int(input("请输入删除的次数: "))
data=[]
for i in range(n): #生成整数列表
    data.append(random.randint(0,20))
print("列表为:", data)
max_s=0
for i in range(② k+1):
    s=add(data[i:n-(k-i)])
    if s>max_s:
        max_s=s
    p=i
print("删除后, 列表为:", ③ data[p:n-(k-p)], "最大值为:", max_s)
```

(1) ①处需要实现对列表中元素的累加, 故填入代码为  $s=s+i$ 。②处的循环范围是删除次数的所有可能, 应遍历到  $k+1$ 。③处需要输出删除后的列表截取部分,  $p$  是开始位置,  $n-(k-p)$  是结束位置, 故填入代码为  $data[p:n-(k-p)]$ 。(2) 程序通过遍历所有可能的删除方式来计算最大值, 这是枚举算法的应用。



(2) 该程序段采用的算法是 **B** (单选, 填字母: A. 解析算法 / **B** 枚举算法)。

14. 编写Python程序，判断两个字符串是否相等：规定字符“?”为万能字符，即可与任意一个字符相等，在忽略字符串中空格以及不区分大小写的前提下，判断两个字符串是否相同。程序运行界面如图所示，请回答下列问题：

```
请输入一个字符串： ?? a d ??dad wd
请输入另一个字符串： a? a? c?d?d w?
两个字符串相同
```

(1) 根据上述规则，字符串“?? a d ??dad wd”和字符串“a? ?? c?d?d ?d”  
相同（选填：相同 / 不相同）。

(2) 实现上述功能的Python程序如下,请在划线处填入合适的代码。

```
s1=input("请输入一个字符串:")
s2=input("请输入另一个字符串:")
#upper将小写字母转换为大写字母
s1=s1.upper()
s2=s2.upper()
s=""
#将字符串s1中的空格去掉
for i in s1:
    if i!=" ":
        _____① s=s+i
    s1=s
#同上,将字符串s2中的空格去掉,代码略
i=0
```

```
if len(s1)!=len(s2):
    print("两个字符串不相同")
else:
    while i<len(s1):
        c1=s1[i];c2=s2[i]
        if c1==c2:
            _____② i=i+1 或 i+=1
        else:
            if _____③ c1=="?" or c2=="?"
                i+=1
            else:
                break
    if i==len(s1):
        print("两个字符串相同")
    else:
        print("两个字符串不相同")
```

(1) 两个字符串去除空格后分别为"??ad??dadwd"和"a???c?d?d?d", 按照规则进行比较, 两个字符串相同。 (2) ①处遍历字符串s1, 将非空格的元素逐一连接到字符串s中, 故填入代码为s=s+i。 ②处用变量i控制while循环的遍历, 若c1和c2中的字符相等, 则变量i加1, 继续比较下一个元素, 故填入代码为i=i+1或i+=1。 ③处是c1和c2不相同的情况, 此时用万能字符“?”进行比较, 若c1和c2的其中一个是“?”, 则判定字符相同, 故填入代码为c1=="?" or c2=="?"。

15. [2025强基联盟] 卡普雷卡尔数 (Kaprekar) 是一种特殊的数字, 其定义为: 一个大于等于4的正整数, 当它的平方被分割成两部分 (分割点可以是任何位置) 时, 这两部分之和等于原来的数。例如, 297 是一个卡普雷卡尔数, 因为  $297^2=88209$ , 而  $88+209=297$ 。小明编写Python程序判断输入的非负整数是否为卡普雷卡尔数, 程序运行效果如图所示。

请输入一个大于等于4的正整数: 297  
297 是一个卡普雷卡尔数

请回答下列问题:

(1) 根据题中卡普雷卡尔数的定义, 9   A   (单选, 填字母: A. 是 / B. 不是) 卡普雷卡尔数。

(2) 实现上述功能的Python程序如下,请在划线处填入合适的代码。

```
n=int(input("请输入一个大于等于4的正整数:"))
```

①  $m=n**2$  或  $m=n*n$

```
a=[];i=10
```

```
while i<=m:
```

```
#append方法用于在列表末尾添加新元素
```

```
a.append(m%i)
```

```
a.append(m//i)
```

②  $i*=10$  或  $i=i*10$

```
for i in range(0,len(a),2):
```

```
if ③  $a[i]+a[i+1]==n$ 
```

```
print(n,"是一个卡普雷卡尔数")
```

```
break
```

```
else:
```

```
print(n,"不是一个卡普雷卡尔数")
```

(1) 9的平方为81,81分割成8和1,8+1=9,故9是卡普雷卡尔数。(2) ① 计算正整数n的平方数,故填入代码为 $m=n**2$ 或 $m=n*n$ 。② 分割点可以是任何位置,通过while循环将所有分割结果存储在列表a中,故填入代码为 $i*=10$ 或 $i=i*10$ 。③ while循环每次将分割的左半部分和右半部分分别存储在 $a[i]$ 和 $a[i+1]$ 中,若两部分之和等于原来的数,则原来的数为卡普雷卡尔数,故填入代码为 $a[i]+a[i+1]==n$ 。

16. [2025山海协作体] 找零问题。假设现有1分、2分、5分、1角、2角、5角、1元面值的硬币。在超市结账时，如果需要找零钱，收银员希望将最少的硬币数找给顾客。给定需要找的零钱数目，如何求出最少的硬币个数？程序运行效果如图所示。

```
请输入需要找的零钱：6.52  
用了 6 个面值为 1 的硬币  
用了 1 个面值为 0.5 的硬币  
用了 1 个面值为 0.02 的硬币  
找零完成!
```

请回答下列问题：

(1) 实现上述功能的Python程序如下，请在划线处填入合适的代码。

```
d=[0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1] #硬币面值
d_num=[20, 20, 20, 20, 20, 20, 20] #现存硬币数量
money=0
for i in range(len(d)):
    money+=_____① d[i]*d_num[i]
    sum=float(input("请输入需要找的零钱："))
    if _____② money<sum
        print("零钱不够，无法找零！")
    else:
        for i in range(len(d)) : #改错
            _____③ range(len(d)-1,-1,-1)
            _____  

            或其他等价答案
```

```
if sum>=d[i]: n=int(sum/d[i])
    _____③ n=sum//d[i]
    if n>=d_num[i]:
        n=d_num[i]
    sum=round(sum-n*d[i], 2)
    print("用了", n, "个面值为", d[i], "
的硬币")
    if sum==0: #sum:剩余要找零金额
        print("找零完成！")
    else:
        print("找零失败！")
(2) 程序加框处代码有错，请改正。
(3) 若需要找的零钱数目为4.92，则最少的硬
币总个数为_____8_____。
```

(1) ①`money`存储所有硬币面值的总和，故填入代码为`d[i]*d_num[i]`。②若所有硬币面值的总和小于需要找的零钱，则输出“零钱不够，无法找零！”，故填入代码为`money<sum`。③若`sum`大于等于`d[i]`，则可以使用面值为`d[i]`的硬币找零，`n`计算使用该硬币的数量，故填入代码为`n=int(sum/d[i])`。(2)为了计算最少的硬币个数，应从面值最大的硬币开始找，即从列表`d`的末尾元素开始，往左遍历，故正确代码为`range(len(d)-1,-1,-1)`或其他等价答案。(3)若需要找的零钱数为4.92，则用4个面值为1的硬币、1个面值为0.5的硬币、2个面值为0.2的硬币、1个面值为0.02的硬币，最少为8个硬币。

17. [2022. 7浙江]查找与替换。从键盘上分别输入要查找和替换的字符串，对文本文件进行查找与替换，替换后保存到新的文本文件中。

完成查找与替换功能的思路是：首先可从待检索文本文件“in.txt”逐行读取文本内容到列表text，然后从键盘上输入查找的字符串key和替换的字符串new，对列表text中的元素逐个进行查找并替换，结果保存到列表result，最后将result写入文件“out.txt”。

(1) 主程序。

```
text=readfile("in.txt") #读入文件
key=input("请输入要查找的字符串：")
new=input("请输入要替换的字符串：")
result=[]
for line in text:
    newline=replace(key,new,line) #替换
    result.append(newline) #添加到列表
writefile("out.txt",result) #写入文件
```

该程序段采用的算法是 **B** (单选，填字母：A. 解析算法 / **B** 枚举算法)。

(2) 读写文本文件。如下的readfile函数，逐行读取文本文件数据存入列表并返回。请在划线处填入合适的代码。

```
def readfile(filename):  
    #打开文件  
    f=open(filename, encoding="UTF-8")  
    text=[]  
    line=f.readline() #从文件中读取一行  
    while line:  
        text.append(line) #添加到列表  
        line=f.readline()  
    f.close()  
    return           text            
  
def writefile(filename, text):  
    #将text写入filename文件，代码略
```

(3) 查找字符串。如下的findstr函数，在字符串line中从begin位置开始查找key在字符串line中的位置，请在划线处填入合适的代码。

```
def findstr(key, line, begin):  
    for i in range(begin, len(line)-len(key)+1):  
        if line[i:i+len(key)]==key:  
            return i  
    return -1
```

(4) 替换字符串。如下的replace函数，在字符串line中检索所有的字符串key并替换为new，请在划线处填入合适的代码。

```
def replace(key, new, line):  
    begin=0  
    while begin<len(line)-len(key)+1:  
        pos=findstr(key, line, begin)  
        if pos==-1:  
            break 或其他等价答案  
        else:  
            line=line[0:pos]+new+line[pos+len(key):len(line)]  
            begin=pos+len(key)  
    return line
```

(1) 解析算法的基本思想是指根据问题的前提条件与所求结果之间的关系，找出求解问题的数学表达式，并通过表达式的计算来实现问题的求解，枚举算法的基本思想是把问题所有可能的解一一列举，然后判断每一个列举出的可能解是否为正确的解。从题干“……逐行读取文本内容到列表text……”可以推断出使用了枚举算法。

(2) 根据题干“逐行读取文本文件数据存入列表并返回”，观察列表相关语句“text=[]”和“text.append(line)”，应填入的代码为text。

(3) 自定义函数findstr的功能是在字符串line中查找key，如果找到了，则返回第一次出现key的索引，如果不存在key，则返回-1。从语句“for i in range(begin, len(line)-len(key)+1)”看出for循环语句遍历索引，因此从line中选择从i开始长度为len(key)的字符，和key进行比较，故填入代码line[i:i+len(key)]==key。

(4) 自定义函数replace的功能是在字符串line中检索所有的字符串key并替换为new，程序使用了while循环从begin开始遍历字符串line，调用自定义函数findstr在字符串line中查找key，返回值为pos，若pos==-1成立，说明当前行不存在key，直接跳出循环或终止循环即可，故填入代码为break，或其他能跳出循环的、改变变量begin的语句。

18. [2023. 6浙江选考]某仓库有一排连续相邻的货位，编号依次为 $0 \sim n-1$ ，用于放置A、B两种类型的箱子，A型箱子占2个相邻货位，B型箱子占1个货位。编写程序，根据已完成的放置或搬离操作，输出空货位数及还可放置A型箱子的最多数目（不移动已放置的箱子）。请回答下列问题：

(1) 若 $n$ 为10，开始时货位全空，经过如图所示的放置或搬离操作后，不移动已放置箱子的情况下，还可放置A型箱子的最多数目为 2 个。

箱子类型	操作类型	货位编号
B	放置	5
A	放置	2,3
B	放置	0
A	放置	7,8
A	搬离	2,3

	0	1	2	3	4	5	6	7	8	9
放置			■	■		■		■	■	
搬离						■		■	■	

(2) 实现上述功能的部分Python程序如下,请在划线处填入合适的代码。

#读取货位总数,存入n,代码略

cnt1=n

lst=[0]\*n #货位状态,0表示对应的货位为空

while True:

'''读取本次已操作的数据:箱子类型、操作类型、货位编号起始值,存入t、d和s,代码略'''

if t=='A':

w=2

\_\_\_\_\_① elif t=='B'

w=1

else: #t不是'A'或'B'时退出循环

break

#d为'P'时表示放置,否则表示搬离

if d=='P': #放置

\_\_\_\_\_② cnt1-=w 或 cnt1=cnt1-w

else: #搬离

cnt1+=w

lst[s]=1-lst[s]

if t=='A':

lst[s+1]=1-lst[s+1]

i,cnt2=0,0

while i<n-1: #统计放置A的数量

if lst[i]==0 and lst[i+1]==0:

\_\_\_\_\_③ i+=1 或 i=i+1

cnt2+=1

i+=1

print('当前空货位数:',cnt1,'还可放置A型箱子的最多数量:',cnt2)

(1) 10个空位放置情况如图所示，一个A型箱要占2个相邻货位，故最多可放2个。

	0	1	2	3	4	5	6	7	8	9
放置										
搬离										

(2) ①处从 $cnt1=n$ 可以看出， $cnt1$ 是空位数量。搬离时 $cnt1+=w$ ，说明变量 $w$ 为应搬离的数量，那么当 $t=='B'$ 时，搬离数量 $w$ 应为1，故填入代码为`elif t=='B'`。②处搬离时空位加 $w$ ，则放置时空位减 $w$ ，故填入代码为 $cnt1-=w$ 。③处列表`lst`存储货位状态，0表示对应的货位为空，故此处统计连续两个空位的个数并存储到变量 $cnt2$ ，统计完后索引 $i$ 的值加2，由于`if`结构下面索引 $i$ 已经加1，故填入代码为 $i+=1$ 。

19. [2024.6浙江选考] 某监控设备可定时采集红绿灯信号状态数据，数据格式记为 $[a, b]$ ，其中 $a$ 、 $b$ 分别为红灯和绿灯的状态值，0表示灯灭，1表示灯亮，如 $[0, 1]$ 表示红灯灭、绿灯亮。

现要编写程序，每隔1秒采集并检测信号灯是否存在如下异常状态：第一类，红绿灯同亮或同灭；第二类，红灯或绿灯超时，即保持同一状态时长大于上限值（如300秒）。检测到异常状态就发送相应信息。请回答下列问题：

(1) 若检测到“红绿灯同亮”异常，则采集到的数据是\_\_\_\_\_ **D** \_\_\_\_\_（单选，填字母）。

A.  $[0, 0]$

B.  $[0, 1]$

C.  $[1, 0]$

D.  $[1, 1]$

(2) 实现上述功能的部分Python程序如下, 请在划线处填入合适的代码。

#设置信号灯保持同一状态时长上限值

tlimit=300

pre=[-1, -1]

''' t[0]、t[1]分别记录红灯、绿灯保持同一状态的时长'''

t=[0, 0]

while True:

''' 接收一次采集到的状态数据, 存入d, 代码略'''

if \_\_\_\_\_ ① **d[0]==d[1]**

if d[0]==1:

#发送“红绿灯同亮”信息, 代码略

else:

#发送“红绿灯同灭”信息, 代码略

for i in \_\_\_\_\_ ② **range(2)**  
if d[i]==pre[i]: **或 range(len(d))**

t[i]+=1

if \_\_\_\_\_ ③ **t[i]>tlimit**

if i==0:

''' 发送“红灯超时”信息, 代码略'''

else:

''' 发送“绿灯超时”信息, 代码略'''

else:

t[i]=1

pre=d

#延时1秒, 代码略

(1) 根据题意, 0表示灯灭, 1表示灯亮, 故红绿灯同亮为[1,1]。 (2) ①根据if下面的语句块可知, 划线处判断红绿灯是否处于相同状态, 故填入代码为`d[0]==d[1]`。 ②根据for循环中的语句可知, i是列表d、pre和t的索引, 故填入代码为`range(2)`。 ③循环遍历两个灯的状态, 若当前状态与上一次状态相同, 则持续时长t[i]加1, 若持续时长大于上限值tlimit, 则发送超时信息, 故填入代码为`t[i]>tlimit`。 +=1。



**谢谢**