Preperation For The Final Test

xaero

Reviewing Material

信息技术学考导引试题详解

First Edition

浙江省杭州高级中學

目录

1 必修 1 · 数据与计算

Part

Sec 1.1 数据与信息

1. 答案：D。考查数据与信息概念的理解，数据、数字的差别。
   A. 数据是对客观事物的符号表示，如图形符号、数字、字母等。在计算机中的表示形式可以是文字、图形、图像、音频、视频等。
   B. 数据可以加过处理，但显然你可以让他失去原有的价值。
   C. 数字放到特定的环境、语境下才有意义，即要有上下文才有含义。
   D. 正确。
2. 答案：B。考查信息概念的理解，信息的特征。
   A. 实验误差是测量值和真实值之间的偏差，不是虚假信息。
   B. 正确。
   C. 同一个信息对于不同的人价值可能不一样。
   D. 信息是信号、消息中所包含的含义，必须依附与数字、文字、图形、图像等载体。
3. 答案：D。考查信息概念的理解，信息的特征。
   A. 互联网上只有已数字化的信息，没有数字化当然查不到。
   B. 知识的获得是人利用自身已有的知识对信息进行加工，进而将新的信息纳入自己的知识结构的过程。检索到也只是看到，并不一定已内化成自己的知识。
   C. 天才也要记单词啊。
   D. 正确。
4. 答案：A。
   A. 。
   B. 。
   C. 。
   D. 。
5. 答案：C。
   A. 若化成十进制计算：$10H = 16D, 10B = 2D, 16D + 2D = 18D$。
   B. $1AH + 2AH = 44H$，注意十六进制下 $A + A$ 等于 14。
   C. 正确。
   D. 若化成十六进制计算：$10D + 10B = AH + 2H = CH$，即十六进制的值是 C。
6. 答案：D。
   A. 。

B. 。

C. 。

D. 。

7. 答案：D。

A. 。

B. 。

C. 。

D. 。

8. 答案：D。

A. 。

B. 。

C. 。

D. 。

9. 答案：B。

A. 。

B. 。

C. 。

D. 。

10. 答案：B。

A. 。

B. 。

C. 。

D. 。

11. 考查信息编码、容量计算。

(1) 视频容量 = 每帧图像容量 × 帧频，每帧图像容量 = 像素点数 × 量化位数。依题意，单张图像容量是：$\frac{1280 \times 720 \times 24}{8 \times 1024 \times 1024} \approx 2.64$MB。因此视频容量是：$2.64 \times 5 \times 60 \times 25 = 19800$MB。压缩比至少是 39.6 : 1 才能压缩到 500MB 以内。答案是 40 : 1。

(2) 加入数据不会改变原来的压缩比，相当于不会改变原先的压缩编码方式（真因为如此，加入音频后的视频容量增加，压缩比不变的话，压缩之后的作品容量也增加，势必会超过 500MB，因此必须重新设定新的、更大的压缩比才能压缩到 500MB 以内，这应该是本题想考查的一个实际应用情景）。

(3) 压缩会使画面不清晰，原因压缩比太大，或者压缩算法太差。改进办法是可以换一种压缩算法（换一个压缩软件），或者保证内容完整的前提下，减少画面尺寸、缩短时长等。

Sec 1.2 算法与程序设计

1. 答案：A。
   A. 。
   B. 。
   C. 。
   D. 。
2. 答案：A。
   A. 实验误差是测量值和真实值之间的偏差，不是虚假信息。
   B. 正确。
   C. 同一个信息对于不同的人价值可能不一样。
   D. 信息是信号、消息中所包含的含义，必须依附与数字、文字、图形、图像等载体。
3. 答案：D。
   A. 互联网上只有已数字化的信息，没有数字化当然查不到。
   B. 知识的获得是人利用自身已有的知识对信息进行加工，进而将新的信息纳入自己的知识结构的过程。检索到也只是看到，并不一定已内化成自己的知识。
   C. 天才也要记单词啊。
   D. 正确。
4. 答案：C。
   A. 。
   B. 。
   C. 。
   D. 。
5. 答案：B。
   A. 若化成十进制计算：$10H = 16D, 10B = 2D, 16D + 2D = 18D$。
   B. $1AH + 2AH = 44H$，注意十六进制下 $A + A$ 等于 14。
   C. 正确。
   D. 若化成十六进制计算：$10D + 10B = AH + 2H = CH$，即十六进制的值是 C。
6. 答案：C。
   A. 。
   B. 。
   C. 。
   D. 。
7. 答案：D。
   A. 。
   B. 。
   C. 。
   D. 。

8. 答案：A。
   A. 。
   B. 。
   C. 。
   D. 。
9. 答案：D。
   A. 。
   B. 。
   C. 。
   D. 。
10. 答案：C。
   A. 。
   B. 。
   C. 。
   D. 。
11. 答案：B。
   A. 。
   B. 。
   C. 。
   D. 。
12. 答案：B。
   A. 。
   B. 。
   C. 。
   D. 。
13. 答案：A。考查 Python 循环语句、双重循环程序的阅读理解。

```
for i in range(1, 7):
    for j in range(1, 7):
        if j <= i:
            print(j, end=" ")
        else:
            print("", end="")
    print()
```

固定第 1 行处的外层循环 i 的值为 1 时，内层循环 j 从 1 变化到 6，对于每一个 j，当 $j \leq i$ 时输出 j 的值，否则输出空值。因此当 $i = 1$ 时，输出 1，然后换行；当 $i = 2$ 时，输出 1 2 然后换行，当 $i = 3$ 时输出 1 2 3 然后换行……，答案选 A。

14. 答案：
15. 答案：
16. 答案：
17. 答案：
18. 答案：
19. 答案：

20. 考查应用 Python 程序解决实际问题的能力。考查字符串的处理与应用。

(1) 考查题意的理解，这是理解题目情景的关键。“we put the bed in the bedroom”中有两处“bed”，会被替换两次。

(2) 阅读与推导过程:

```
1  text = input("输入原文字符串: ")
2  key = input("输入要查找的字符串: ")
3  rs = input("输入替换字符串:")
4  result = "";  count = 0;  i = 0;  n = len(text)
5  while i < n - len(key) + 1:
6   s = text[          ①          ]
7   if key == s:
8       result += rs
9       count += 1
10      i += len(key)
11  else:
12      result += text[i]
13      i += 1
14          ②
15 if count > 0:
16  print("替换的次数: ", count)
17  print("替换后的结果: ", result)
18 else:
19  print("要查找的内容不存在")
```

(*a*) 第 5 行的循环和 n 有关，而 n 是原文的长度，因此第 5 行的循环是在扫描原文的每个字符。

(*b*) 从 i 的变化上看，当第 8 行两个字符相等时，i 往后移动与 key 一样的长度；当两个字符不等时，i 往后移动 1 个字符长度，所以 i 是指示了原文 text 中待比较字符串的索引位置信息。

(*c*) 循环中第 7 行判定了 key 是否与 s 相等，那么 s 就需要从原文中截取一个字符串，再与 key 作相等比较，因此第①空应该填写原文字符串的切片，切片的起始值是当前 i 的值，切片的长度应该与 key 的长度相等，于是第①空答案是 `i:i+len(key)`。

(*d*) 在解题时一定要用样例带入后阅读，比如原文 `text="Abedrbedom"`，待替换字符串 `key="bed"`，那么当 $i=1,5$ 时分别找到两处“bed”，如下图所示。当 i 指向 8 号位置时，剩余字符串已不足 3 位（即待查找值 key 的长度），也就无需继续循环，这也是第 5 行 while 循环条件是 `i < n - len(key)+ 1`而不是 `i < n`的原因。但是这样带来的后果是剩余的字符串无法原样连接到 result 中取，如下图中的最后两个字符“om”。因此需要在循环结束时，第 14 处将剩余字符串连接到最终结果上。第②空的答案是 `result += text[i:]`，其中切片的终止端点写明 n 亦可。

```
0 1 2 3 4 5 6 7 8 9
A b e d r b e d o m
  ↑
  i=1时找到第一处，字符串替换后，i=i+3，指向4号位置
------------------------------------------------------------------
0 1 2 3 4 5 6 7 8 9
A b e d r b e d o m
          ↑
```

```
i=5时找到第二处，字符串替换后，i=i+3，指向8号位置
```

21. 考查应用 Python 程序解决实际问题的能力。考查字符串的处理与应用。

(1) “Good”中的四个字母来自键盘不同的两行按键，故答案是 No。

(2) 阅读与推导过程：

```
1  def to_lower(ch):          # 转小写字母
2      if ch >= "A" and ch <= "Z":
3          return chr(ord(ch) + 32)
4      else:
5                  ①
6  line_1 = "qwertyuiop"    # 键盘第一行字母
7  line_2 = "asdfghjkl"     # 键盘第二行字母
8  line_3 = "zxcvbnm"       # 键盘第三行字母
9  char = input()
10 c1 = 0
11 c2 = 0
12 c3 = 0
13 n = len(char)
14 for ch in char:
15     ch =            ②
16     if ch in line_1:
17         c1 += 1
18     elif ch in line_2:
19         c2 += 2
20     elif ch in line_3:
21         c3 += 3
22 if c1 == n or c2 == n * 2 or c3 == n * 3:
23     print("yes")
24 else:
25     print("no")
```

(*a*) 第①空容易填：第 2 行 if 语句判定了大写字母，第 3 行将其转成小写字母，因此非大写字母时直接返回 ch，答案是 `return ch`。

(*b*) 第 14 行循环语句遍历提取了输入字符串 char 中的每个字符，在第 16、18、20 行分别判定了是否是键盘上哪一行的字母：第一行则 $c1$ 加 1，第二行则 $c2$ 加 2，第三行则 $c3$ 加 3，由此可以断定，在第②空处需要将字母规范化——统一转成小写字母，这就需要调用 `to_lower()`函数，因此答案是 `to_lower(ch)`，参数是当前扫描到的字符 ch。

(*c*) 对于第 22 行条件的理解：如果输入字符 char 都来自键盘第一行，那么 $c1$ 的值与 char 字符串长度相等，因此 $c1$ 每次都加 1；同理，如果都来自第二行，则 $c2$ 的值是 char 字符串长度的两倍，因为 $c2$ 每次都加 2；$c3$ 的值亦同理。

(3) `c1 += 1`的含义是第一行的字符数量增加 1

22. 考查应用 Python 程序解决实际问题的能力。考查随机数函数、枚举算法。

```
1  import random
2  n = int(input("请输入要产生的英文字符串长度："))
3  s = ""
4  for i in range(n):
5      # randint(1,58): 随机生成一个 [1,58] 范围内整数，字母 A 的 ASCII 码值为 65
6      s += chr(64 + random.randint(1, 58))
```

```
print(s)
ans = input("请按样例输入: ")
c = 0
for i in range(n):
    if          ①          :
        c += 1
p =          ②
print("正确数量: ", c, ", 正确率为: ", p, "%")
```

(*a*) 判断两个字符串有多少个字符相同，可以用枚举算法：遍历每个字符串的每一位，分别判定是否相等。

(*b*) 由第 10 行的 for 循环语句知，i 取遍了 $[0,-1]$ 的每个数，这相当于字符串的索引值。而第 11 行处的条件成立时，变量 c 的值加 1，又由第 14 行的输出可知 c 是正确单词的个数。因此第 ① 空是判定两个字母是否相等，原始字符串是 s，用户输入字符串是 ans，因此答案是 `s[i] == ans[i]`。

(*c*) 变量 p 是什么？同样可以看 14 行的输出语句——p 是正确率百分比。因此 p 的计算方式是正确个数除上总个数，答案可以是 `c / n * 100`。题意没有说如何保留小数，也没有输出示例，因此这个答案也可以。参考答案是 `int(c / n * 100 + 0.5)`，它的功能是四舍五入保留整数。

23. 考查应用 Python 程序解决实际问题的能力。考查进制转换解析算法、字符串应用。

```
def conv(s):
    ans = ""
    if s > "9":
                ①
    else:
        s = int(s)
    while s > 0:
        k = s % 2
        s //= 2
                ②
    for i in range(4 - len(ans)):
        ans = "0" + ans
    return ans
s = "2A08:CCD6:0088:108A:0011:0002:202F:AA05"
ans = ""
flag = False
for i in s:
    if i == ":":
                ③
        ans += i
    elif i != "0" or flag == True :
                ④
        flag = True
print("原IPv6地址为:", s)
print("去前导零后:", ans)
```

(*a*) 当自定义函数比较复杂时，可以从主程序开始阅读。那么从第 14 行开始阅读程序：第 17 行遍历取出了字符串 s 中的每个字符，s 是个十进制模式的 IPv6 地址字符串，依题意需要将它转成二进制模式，可以猜测本题处理思路就是逐个取出 s 的每个字符并转

换成二进制并输出结果。

(*b*) 第 18 行是 *i* 的值是冒号的情况，这意味着冒号前面一段 IPv6 已转换完，*i* 中的值直接连接到最后结果字符串 ans 变量的后面，ans 变量的功能也还是从最后一行的输出语句中得到。但是第 ③ 空还不知道填什么。

(*c*) 第 21 行是说当 *i* 不是 0，或者 flag 的值是 True 时执行第 22、23 行程序。容易相当，当 *i* 非 0 时必然要转成二进制格式，因此这里需要调用前面的 conv() 函数，调用结果应该是 *i* 字符对应的二进制串，同样要把二进制串连接到 ans 变量后面，于是第 ④ 空的答案应该是 `ans += conv(i)`，其中 *i* 就是当前需要转换成二进制的十六进制字符。

(*d*) 从第 21 行的 elif 判定结果看，当 $i \neq 0$ 时，flag 的结果会变成 True，一个隐含的情况是当 $i = 0$ 但 flag 值是 True 的时候，也会执行 21、22 两行代码，即也会将该“0”转成二进制串。结合题意“前导零可以省略”可知，非前导（中间的）零需要转换。由此断定 flag 的值为 True 表示当前有非前导的零（需要转换）；flag 的值为 False 时，若出现零则是前导的零。再结合第 16 行 f 初值 False 可知，这样的推论是合适的。

(*e*) 因此，第 ③ 空是出现冒号后，下一次得到的字符串若是零，该零必然是前导的零，于是这里填 `flag = False`。

(*f*) 转到 conv() 函数，可以断定参数变量 *s* 是待转换的十六进制字符串，这在第 8、9 两行的循环模 2 取余也可以得到验证（转二进制的方法就是除二取余法）。

(*g*) 但是 *s* 是十六进制，除二取余之前需要转成十进制。阅读第 1 ～ 6 行的 if 语句可以看出，若 $s \leq 9$，则直接取整（此时十六进制值与十进制相等）；否则要把“A”转成 10，把“B”转成 11……把“F”转成 15。把字母转整数可以用 ASCII 码函数 ord()，本题答案是 `ord(s)- 65 + 10`。

(*h*) 第 ② 空是将余数 *k* 连接成二进制串的语句，注意最先除二取余得到的余数是最低位，最后得到的余数是最高位，因此本题答案需要将 *k* 转成字符串后连接到答案变量 ans 的前面。本题填 `ans = str(k)+ ans`。

24. 考查应用 Python 程序解决实际问题的能力，考查列表的应用。

```
import random as rd
data = [180,283,385,170,276,384,180,285,380,190,295,390,170,272,372]
s = [0, 0, 0]                          # 存储 3 个作品的得分
ans = []                               # 存储并列最高平均分的作品号
maxb = 0
for i in range(len(data)):
    zp = ____①____                     # 分离出作品编号
    fs = data[i] % 100
    ____②____                          # 累加当前作品的得分
for j in range(3):
    ____③____
    print("作品", j+1, "平均分为", s[j])
    if s[j] > maxb:
        maxb = s[j]
for z in range(3):                     # 查找并列最高平均分
    if ____④____:
        ans.append(z + 1)              # 将数据添加到列表 ans 尾部
print("平均分最高作品号是: ", ans)
```

(*a*) 由第 6 行的循环范围知，该 for 循环遍历了 data 列表的每个元素，*i* 是其索引值。

(*b*) 分离字符串可以用切片，分离整数的各个数位可以用对十取余数，或者整除十。从循环中的第 8 行知，`data[i] % 100`就是该整数的十位和各位上的数，即第 *i* 号数据中的得分值。第 ② 空，分离百位数可以模仿这写 `data[i] // 100`，除以 100 之后的整数部分就是作品号。

(*c*) 按注释，第 ② 空应是累加 fs 的值到列表 *s* 中的某个位置上，这个位置应该与作品编号有关。注意到第 7 行分离出来的作品编号都是从“1”开始数的，而 *s* 的索引值是从“0”开始计的，因此需要修正，答案是 `s[zp-1] += fs`。

(*d*) 因为有 3 个作品，因此第 10 行的 3 次循环应该遍历了每个作品，并输出了相应的平均分。也就是说，第 12 行输出的 s[j] 是第 *j* 号作品的平均分，而之前 *s* 中保存的是每个作品的总分，因此第 ③ 空需要求平均分，答案是 `s[j] /= 5`。

(*e*) 第 13 行是打擂算法：若当前平均分 s[j] 大于“擂台”上的数 maxb，则让 s[j] 留在擂台上。所以 maxb 保存的是最大值。

(*f*) 第 ④ 空的条件成立时将 *z* 存入列表 ans，而由最后一行的输出结合输出图示看，列表 ans 中存放了所有得分都是最高的作品编号。第 ④ 空就得填写某个作品均值与 maxb 相等时执行插入操作，答案是 `s[z] == maxb`。

Sec 1.3 数据处理与应用

1. 答案：D。考查数据整理方法与目的。
2. 答案：C。
   A. 实验误差是测量值和真实值之间的偏差，不是虚假信息。
   B. 正确。
   C. 同一个信息对于不同的人价值可能不一样。
   D. 信息是信号、消息中所包含的含义，必须依附与数字、文字、图形、图像等载体。
3. 答案：B。
   A. 互联网上只有已数字化的信息，没有数字化当然查不到。
   B. 知识的获得是人利用自身已有的知识对信息进行加工，进而将新的信息纳入自己的知识结构的过程。检索到也只是看到，并不一定已内化成自己的知识。
   C. 天才也要记单词啊。
   D. 正确。
4. 答案：B。考查 pandas 数据处理 drop() 函数、groupby() 函数功能的理解。注释如下：

```
import pandas as pd                                  # 导入并使用 pd 作为别名
df = pd.read_csv("mnxk.csv", sep=",")                # 读取数据
df1 = df.drop("已选科目数", axis=1)                    # 删除“已选科目数”列
print(df.head())                                     # 打印 df 的前 5 行
print(df1.head())                                    # 打印 df1 的前 5 行
sc=df1.groupby("班级", as_index=False).count()       # 按“班级”分组
m = len(df)                                          # df 的行数
n = len(df1)                                         # df1 的行数
print(sc)                                            # 分组后的数据
```

注意 pandas 的很多操作处理后原始数据都不会改变。比如，第 3 行 drop() 函数删除了“已选科目数”列，参数“axis=1”指明了这是列而不是行。该函数调用后，产生了一个新的数据集合并赋值给对象 df1，而原始的数据集合 df 未曾变化。选项 B 就考查了 pandas 数据处理的这个特点：第 4 行打印的结果是原始数据的前 5 行，包含“已选科目数”这列数据，而第 5 行的输出的 5 行数据虽然大部分与前面相同，但不含“已选科目数”这列数据。选项 C 考查的是 df 和 df1 数据对象的行数是否相同，由于没有删除行，行数必然是一样的。选项 D，第 6 行的分组可以让相同班级的数据合并成一行数据，这个数据的每列数据是原先该列数据的非空单元格个个数（即 count() 函数的功能）。如，若原始数据如左侧所示，则执行第 6 行分组语句后的结果如右侧所示。在右侧数据中，“1 班”的“Name”值是 4，表示原始数据中 1 班“Name”列数据非空单元格个有 4 个；“1 班”的“物理”值是 2，表示原始数据中 1 班“物理”列数据非空单元格个有 2 个（相当于 1 班有两个 2 人选了物理）。

	班级	Name	物理	历史	技术	化学
0	1班	张三丰	1	1	1	
1	2班	郭靖	1		1	
2	1班	小龙女	1			1
3	2班	李秋水				
4	3班	杨过	1		1	1
5	1班	令狐冲				
6	3班	任我行	1	1		1
7	1班	黄蓉			1	1

	班级	Name	物理	历史	技术	化学
0	1班	4	2	1	2	2
1	2班	2	1	0	1	0
2	3班	2	2	1	1	2

5. 答案：D。

   A. 若化成十进制计算：$10H = 16D, 10B = 2D, 16D + 2D = 18D$。

   B. $1AH + 2AH = 44H$，注意十六进制下 $A + A$ 等于 14。

   C. 正确。

   D. 若化成十六进制计算：$10D + 10B = AH + 2H = CH$，即十六进制的值是 C。

6. 答案：B。

   A. 。

   B. 。

   C. 。

   D. 。

7. 答案：B。

   A. 。

   B. 。

   C. 。

   D. 。

8. 答案：D。

   A. 。

   B. 。

   C. 。

D. 。

9. 答案：C。

A. 。

B. 。

C. 。

D. 。

10. 考查 pandas 数据处理与应用。

(1) 考查数据处理的实际用途，帮助理解题目情景。

(2) 考查 pandas 数据格式的识别。

```
1  def s_review(c):
2      for r in range(df.shape[0]):                  # 批阅 1 个单选题
3          if df.at[r, qnum[c]] == _____①_____:
4              tmp = 3
5              df.at[r, score[c-2]] = tmp
6              df.at[r, score[10]] += tmp            # 计算总分，存入"sum" 列
7  qnum = df.columns
8  sans = "BDCABDDBCB"                               # 本次作业的标准答案
9  score=["sc1","sc2","sc3","sc4","sc5","sc6","sc7","sc8","sc9","sc10","sum"
10 for c in score:
11     df[c] = 0
12 for c in range(2,12):                             # 逐题批阅
13         _____②_____
14 print(df)
15 df.to_excel("客观题成绩.xlsx", index=False)  # 保存结果
```

(3) 解题过程：

(*a*) 从第 7 行主程序开始阅读，对 pandas 程序阅读，一定要直到变量保存的数据是什么？数据的结构是怎样的？

(*b*) 第 7 行 qnum 保存了数据对象 df 的所有列名称，即 `["name", "snum", "ans1", "ans2", ..., "ans10"]`。第 10 行的循环，结合第 9 行 score 列表中的数据可以知道第 11 行在数据对象 df 中新增了很多数据列，列名称分别是“sc1”、“sc2”、“sc3”、……、“sum”，每列的值都是 0。这也是 pandas 的特点，数据列直接可以参与算术运算、关系运算和赋值操作，每种操作都可以将该列的所有行都进行相应处理。

(*c*) 第 ② 空处所进行的循环是逐题批阅。原始表格数据中一题就是一列数据，列序号是 2 ~ 12，刚好能对上这里的循环范围。因此，第 12 行的循环变量 *c* 相当于列序号——不过，pandas 需要的是列名称，这就需要 qnum 中对应的列名称来引用原始数据了。这里需要调用第一行的 s_review() 函数。

(*d*) 阅读 s_review() 函数。第 2 行 df.shape 可以返回数据对象 df 的维度“形状”：行数（df.shape[0]）和列数（df.shape[1]），因此 *r* 就是行索引号。由 `df.at[r, qnum[c]]`操作可知 qnum[c] 必然是列名称，结合前面的分析可以知道 *c* 必然是列序号。由于列表 qnum 中索引 2 号的列名称才是第一题名称“ans1”，因此 *c* 期望的值也应该从 2 开始。那么第 ② 空的函数调用就好办了：函数名已知的，参数作用也推知了，所以答案应该是 `s_reviewww(c)`，就让 *c* 的值从 2 开始传递、调用函数。另外

从 s_review() 函数的结构上看，它有 return 语句返回值，所以这空也无需考虑赋值——直接调用即可。

(*e*) 再回到第 3 行程序，`df.at[r, qnum[c]]`取得了 *c* 列每个人填写的答案，它们需要与标准答案做比较，标准答案保存再 sans 字符串中，它的索引号是从 0 开始的，所以第 ① 空的答案是 `sans[c-2]`。

(*f*) 第 5 行的程序是将该行（第 *r* 行）对应的得分列赋值为 tmp 分分值（如“ans1”列对应的分值是“sc1”列）。第 6 行的程序是将该分值累加到它的总分中去（即“sum”列，它的值是 10 个选择题的得分累加而来）。

11. 考查文本数据处理、分词、字符串统计与字典的应用。

```
1  import jieba                                          # 导入 jieba 模块
2  import pandas as pd
3  text = open("news.txt", encoding="utf-8").read() # 打开文本文件
4  words = jieba.lcut(text, cut_all=False)           # 分词
5  counts = {}
6  for name in words:
7      if len(name) != 1 and not ("a" < name[0] < "z") and not ("0" <
           name[0] < "9"):
8          if name in counts:
9              counts[name] += 1                     # 词语已出现过
10         else:
11             counts[name] = 1                      # 词语第一次出现
12 # 字典转化为 DataFrame 格式存储
13 df = pd.DataFrame(list(counts.items()), columns=["词", "次数"])
14 df = df.sort_values("次数", ascending=False)     # 按“次数”降序排序
15 print(df)
```

(1) jieba 是目前常用的分词模块，它是一个基于词典分词的模块。模块导入后，程序再第 3 行通过 python 的内置 open() 函数打开了文本文件，read() 函数可以读取文件中的所有数据。第 4 行调用了 jieba 的 lcut() 函数进行分词，函数名中的“l”表示分词结果数据是一个列表（即 list，这里了解即可，无需记忆），函数的 cut_all 参数设定为 False 表示是精准分词，不会分隔“词中词”，当该参数设定为 True 时表示全模式分词，会分隔所有词。如“中华人民共和国”，False 模式下结果是一个词 `["中华人民共和国"]`，True 模式下会有多个词 `["中华", "人民", "共和国", "中华人民共和国"]`。对于 words 列表中的每个词，第 7 行程序过滤掉了单字、字母开头的、数字开头的字符串，因此答案选 C。

(2) 本小题考查 jieba 分词的规则特点，因为它是用现有的词典进行分词的，因此想要添加一个新词时,只需在分词前添加该词再进行分词即可。具体可以通过 `jieba.add_word("公益活动")`来添加该词。

(3) 第 8 行程序先判定某个单词 name 是否在字典 counts 的键名中出现过，如果出现过，则直接根据该键名取出其键值，然后加 1 后仍然存放在该键名上。else 分支就是该键名第一次出现，该键值初始为 1，答案是 `counts[name] = 1`。

12. 考查 pandas 数据处理与应用，matplotlib 数据可视化。给出三行数据示例如下：

```
109, 2007-02-20 00:07:10, 121.443100, 31.273000,  0, 45, 0
109, 2007-02-20 00:08:06, 121.447600, 31.272000,  6, 22, 1
```

```
109, 2007-02-20 00:09:07, 121.452500, 31.271000, 46, 67, 1
```

```
1 import pandas as pd
2 import matplotlib.pyplot as plt                # 导入 matplotlib 模块
3 plt.rcParams["font.sans-serif"] = ["KaiTi"] # 图表中中文以楷体显示
4 df = pd.read_csv("Taxi_105.txt", sep=",")
5 df.columns = ["xh", "sk", "jd", "wd", "sd", "jj", "zkzt"]
6 df = _____
```

(1) 第 3 行程序设置字体以便显示中文，这行代码了解即可。第 4 行程序读取了 csv 文件并转成 DataFrame 数据对象保存再 df 中。由于原始数据中没有标题行，程序的第 5 行指定了数据各列的标题。从处理结果上看，“速度”、“夹角”列都可以删除，因此删除这两列数据都可以。drop() 函数删除列的语法是 `df.drop("jj", axis=1)`，其中 axis 参数为 1 指明了删除的是列。

```
7  def pickup(rid):
8      # 计算每次上客的停车时长，代码略
9      return t, id1    # 返回停车时长、上客停车时的数据行索引
10 # 计算该出租车当日载客次数
11 ty = []; px = []; py = []; count = 0
12 n = len(df)
13 for row in range(n-1):
14     if df.at[row, "zkzt"]==0 and df.at[row+1,"zkzt"]==1:     # 上客
15         pt, id = pickup(row)
16         ty.append(pt)
17                 ①
           _________
18         px.append(df.at[id, "jd"])
19         py.append(df.at[id, "wd"])
20 print("该出租车当日载客次数为:", count)
21 # 上客平均时长四舍五入取整
22 print("上客平均时长、最大及最小时长(单位:秒):", _____②_____,
      max(ty), min(ty))
```

(2) 第 12 行程序中 n 取得了 df 数据对象的行数，第 13 行循环了 $n-1$ 次，可以确定 row 就是数据对象 df 的行索引值。第 14 行的判定是说当前行载客状态是空，但是下一行的载客状态是有客则表示当前正在上客。由 pickup() 函数的注释和 return 语句可知，第 15 行中的变量 pt 保存了函数的第一个返回值停车时长，id 保存了第二个值上客时的数据行索引。因此下面几行程序也容易理解：列表变量 ty 保存了各次上客前停车时长，列表 px 保存了各次上车时的经度值，py 保存了各次上车时的纬度值。第 ① 空似乎不需要填什么程序。不过从循环结束后的第 20 行程序上看，这里输出了 count 的值，为载客次数，再考查它的初值是零，因此需要在循环内进行次数统计，于是第 ① 空答案是 `count += 1`。第 ② 空是输出上客平均时长，上车次数是 count，这就需要算出上车总时长（接到各个客人时需要空车停留、闲逛的总时长）。这里已经将各次上车时间保存在列表 ty 中，求和只需调用 sum() 函数即可，四舍五入保留整数可以使用 round() 函数，于是答案是 `round(suum(ty)/count)`。注意很多同学的答案写成了 mean(ty)。在 python 内置函数中，只有求和 sum() 函数，求最值 max()/min() 函数，没有求平均值 mean() 函数。在 pandas 中，数据框对象有 mean() 方法，但它的格式是 df.mean()，而且会求出 df 每一列的平均值，结果不是一个整数而是一个 Series。这里不能用该函数。

```
tx = range(count)
plt.figure(1)
plt.title("当日上客时长对比图(单位:秒)")
          ①              # 绘制图表分析当日搭载乘客的上客时长
plt.figure(2)
plt.title("当日上客地点分布图")
          ②              # 绘制图表分析当日搭载乘客的上客地点分布
plt.show()
```

(3) plt 数据可视化最关键的三要素是：图表类型、横纵轴数据。分析左图可以确定是柱形图，那么使用 `plt.bar()`函数。纵轴数据从纵坐标的值与图表标题上可以看出是上客时长，数据应该是列表 ty 中的值。这里的横轴数据不是很明显，图中也未给出坐标轴标签，不过看数据还是可以知道就是一些序号，结合第一行的程序，还是可以确定横轴数据就是列表 tx 中的值，于是答案是 `plt.bar(tx, ty)`。同样的方法可以确定第②空的答案是 `plt.scatter(px, py)`。

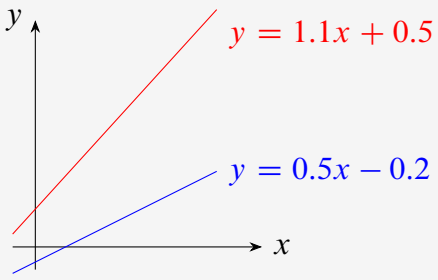
Sec 1.4 人工智能与应用

1. 答案：D。考查数据整理方法与目的。
2. 答案：C。
   A. 实验误差是测量值和真实值之间的偏差，不是虚假信息。
   B. 正确。
   C. 同一个信息对于不同的人价值可能不一样。
   D. 信息是信号、消息中所包含的含义，必须依附与数字、文字、图形、图像等载体。
3. 答案：B。
   A. 互联网上只有已数字化的信息，没有数字化当然查不到。
   B. 知识的获得是人利用自身已有的知识对信息进行加工，进而将新的信息纳入自己的知识结构的过程。检索到也只是看到，并不一定已内化成自己的知识。
   C. 天才也要记单词啊。
   D. 正确。
4. 答案：B。
5. 答案：D。
   A. 若化成十进制计算：$10H = 16D, 10B = 2D, 16D + 2D = 18D$。
   B. $1AH + 2AH = 44H$，注意十六进制下 $A + A$ 等于 14。
   C. 正确。
   D. 若化成十六进制计算：$10D + 10B = AH + 2H = CH$，即十六进制的值是 C。
6. 答案：B。
   A. 。
   B. 。
   C. 。
   D. 。
7. 答案：B。

```
import matplotlib.pyplot as plt
import pandas as pd
# 创建画布和坐标系，此处代码略
df = pd.read_excel("data.xlsx")  # 读取点的坐标值并完成分类存储
x = df["宽度"]
y = df["高度"]
t = df["类别"]
x1 = []; y1 = []; x2 = []; y2 = []
for i in range(          ①          ):
    if t[i] == "柠檬":
        x1.append(x[i]);  y1.append(y[i])
    else:
                  ②
# 绘制散点图
plt.scatter(x1, y1, c="r", marker="*", s=15, label="柠檬")
plt.scatter(x2, y2, c="b", marker="o", s=5, label="苹果")
# 显示图例、设置坐标轴后最后显示散点图。此处代码略
```

(1) 从第 5 ～ 7 行程序看，x, y, t 三个变量分别保存了原始数据的每一列（Series），从第 10 行的 t[i] 使用方式上看，i 就是索引号，因此第 ① 空的范围与数据行数有关，答案是 `len(t)`。第 11 行程序将柠檬的宽高保存到了 $x1, y1$ 中，那么 else 分支应该保存苹果的数据，答案是 `x2.append(x[i]); y2.append(y[i])`。程序第 15、16 行绘制了两个散点图，后面几个参数的功能可以了解一下：参数 c 是 color 的别名，可以绘制散点的色彩，r 就是 red，b 就是 blue；参数 marker 是散点的样式，“*”表示五角星形，“o”表示稍大的圆点；参数 s 是 size 的别名，意味散点的大小；label 就是当调用 plt.legend() 函数时显示的图例中的标签。

```
1  kk = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2,
       1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 3.0, 4.0]
2  bb = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5,
       8, 8.5, 9, 9.5, 10]
3
4  def loss(k, b):
5      n = 0                              # 变量 n 存储总的错误个数
6      for i in range(len(x1)):           # 统计柠檬分类的错误个数
7          if y1[i] < k * x1[i] + b:
8              n += 1
9              ③                          # 统计苹果分类的错误个数
10     return n
11 minloss = 30
12 for k in kk:
13     for b in bb:
14         minloss1 = loss(k, b)
15         if minloss1 < minloss:
16             minloss = minloss1
17             K = k
18             B = b
19 print("求得分类直线的k=", K, "b=", B)
20 # 绘制直线，此处代码略
```



(2) 这题有点数学的味道。对于数学函数 $y = kx + b$，对于不同的 k, b 组合，产生的函数图像是不一样的，如上图所示。因此主程序第 12、13 行用两重循环枚举了不同 k, b 的组合，对于每一组 k, b 组合都使用函数 loss() 计算出位柠檬和苹果分类错误的个数，通过打擂法保留分类错误最小时的 k, b 组合。由此分析，第 ③ 空处的代码于第 6 行的循环类似——第 6 行的循环遍历了列表 $x1, y1$ 中柠檬的宽度和高度值，对于柠檬而言，y 值应该大于 $kx + b$ 的值，因此它用条件 `y1[i] < k * x1[i] + b`来统计错误的数量。苹果可以模仿着写：数据在列表 $x2, y2$ 中，苹果正常的 y 值应该小于 $kx + b$，因此程序可以写成：

```
for i in range(len(x2)):
    if y2[i] > k * x2[i] + b:
        n += 1
```

(3) 可以有两种方法判定：① 将水果宽度 $x = 6.8$ 代入 $y = 0.4x + 5$，得 $y = 7.72$，即分类器计算得到苹果和柠檬的高度分界点是 7.72 厘米，现在水果的高度是 7.3 厘米，小于临界点，应该为苹果。② 将高度值 $y = 7.3$ 代入 $y = 0.4x + 5$，得 $x = 5.75$，即宽度分界点是 5.75 厘米，而该水果得宽度是 6.8 厘米，大于分界点，应该判定为苹果。