

乐清中学2024级高二信息技术 周练(八)

班级：高二（ ）班 姓名：_____

2024.8.8

第 50 题

一、选择题（本大题共 12 小题，每小题 2 分，共 24 分。每小题列出的四个备选项中只有一个是符合题目要求的，不选、多选、错选均不得分。）

1. 以下关于信息的说法，错误的是（ ）

A. 信息必须依附于载体，但载体本身并不就是信息 **画作只是信息的载体，画作的内容才是信息**

B. 信息是可以被再使用的 **信息可以被再使用**

C. 信息是随着时代的发展而变化的

D. 信息也会随着时代发展而变化 **数据也会随着时代发展而变化**

2. 以下关于 UltraEdit 的说法，错误的是（ ）

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
32	30	32	35	2D	D6	DB	C9	BD	B7	E7	F3	DD	C8	FC	21

2025-舟山风筝赛！

以下关于

A. 十六进制数 16 8 0 的十六进制表示是 16 8 0 **字节**

B. 十六进制数 16 8 0 的十六进制表示是 16 8 0 **字节**

C. 十六进制数 16 8 0 的十六进制表示是 16 8 0 **字节**

D. 十六进制数 16 8 0 的十六进制表示是 16 8 0 **字节**

3. 以下关于信息的说法，错误的是（ ）

A. 信息必须依附于载体，但载体本身并不就是信息

B. 信息是可以被再使用的

C. 信息是随着时代的发展而变化的

D. 信息也会随着时代发展而变化

4. 以下关于信息的说法，错误的是（ ）

A. 信息必须依附于载体，但载体本身并不就是信息

B. 信息是可以被再使用的

C. 信息是随着时代的发展而变化的

D. 信息也会随着时代发展而变化

5. 以下关于信息的说法，错误的是（ ）

A. 信息必须依附于载体，但载体本身并不就是信息

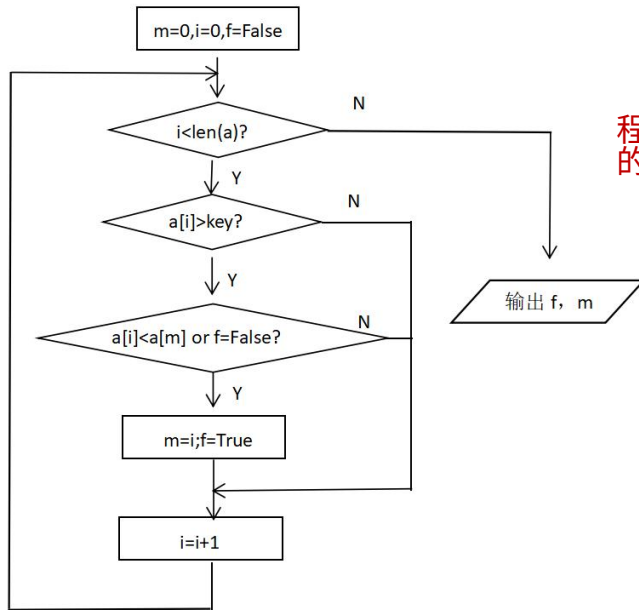
B. 信息是可以被再使用的

C. 信息是随着时代的发展而变化的

D. 信息也会随着时代发展而变化

6. 某算法的部分流程图如图所示，若列表 a 的值为 [23, 37, 55, 45, 33, 42, 89, 70], key 的值为 40，执行这部分流程后，f、m 的值分别为：（ B ）

最小的是42，所在位置是5



程序功能：找到所有大于key的数据里的最小值所在的位置

A. True 3 B. True 5 C. True 4 D. False 0

7. 队列 L 中首到尾的元素依次为 1, 2, 3, 4，栈 Z 初始为空，约定：G 操作为元素出队后入栈，W 操作为元素出栈后入队，经过 GGWGGWGG 系列操作后，栈中从顶到底的元素依次为：（ A ）

A. 4231 B. 2341 C. 1324 D. 4213

8. 有如下 python 程序段：

```

a=[2, 3, 6, 7, 3, 6, 9, 2, 1, 4, 5, 5, 6, 3, 4, 5, ]
max=0
dic={}
for i in a:
    if i not in dic:
        dic[i]=1
    else:
        dic[i]+=1
        if dic[i]>=max:
            max=dic[i]
            v=i
    
```

初始：L:1,2,3,4 Z: 空
 GG → L:3,4 Z:1,2
 W ↓ L:3,4,2 Z:1
 GG ← L:2 Z:1,3,4
 W ← L:2,4 Z:1,3
 GG ← L:1,3,2,4 Z:1,3,2,4

程序功能：统计列表中各类数字出现的次数，找出现次数最多的，如果出现次数最多的有相同的，取后统计的

按程序遍历过程，
 数字2出现2次
 数字3出现3次
 数字6出现3次
 数字7出现1次
 数字9出现1次
 数字1出现1次
 数字4出现2次
 数字5出现3次

数字3、6、5都出现了3次，为出现次数最多的，按程序执行结果，取后统计的5为最终结果

执行该程序段后，max 和 v 的值分别为：（ D ）

A. 3 3 B. 9 1 C. 3 6 D. 3 5

9. BMI 指数是世界卫生组织（WHO）目前常用于判断体重是否标准的简单方法，计算方法为：BMI=体重/身高平方。BMI 值的标准：BMI 低于 18.5 为偏瘦，BMI 介于 18.5 到 25 之间为标准；BMI 高于 25 为超重。小王计算出正确 BMI 值后，编写代码根据 BMI 值判断体重是否标准，下列选项中正确的是（ C ）

A. <pre>s="偏瘦" if BMI>25: s="超重" elif BMI<=25: s="标准"</pre> 会把18.5以下的判定为“标准”	B. 没有=号 <pre>if 18.5<BMI<25: s="标准" elif BMI>25: s="超重" else: s="偏瘦"</pre>	C. <pre>s="标准" if BMI>25: s="超重" elif BMI<18.5: s="偏瘦"</pre>	D. <pre>if BMI>25: s="超重" if 18.5<=BMI<=25: s="标准" else: s="偏瘦"</pre> 会把大于25的判定为“偏瘦”
--	---	---	---

10. 执行下列 Python 程序段后，b[1]的值是（ D ）

```
a=[11, 3, 23, 7, 10, 19]
n=len(a)
b=[0]*n
for i in range(1,n) :
    for j in range(1, n-i+1):
        if a[j]>a[j-1]:
            a[j],a[j-1] = a[j-1],a[j]
            b[i]+=1
```

A. 1 B. 2 C. 3 D. 4

外循环执行一遍，内循环将相邻数据进行比较，把大的换到前面，小的换到后面，一遍完成，最小的会被换到最后面
b数组记录每一遍比较，交换的次数
b[1]记录的是第一遍比较时的交换次数
根据原始数据，第一遍比较过程为：11与3比较，无需交换，3与23比较，交换1次，3与7比较，交换1次，3与10比较，交换1次，3与19比较，交换1次，完成比较，交换了4次

11. 若字符 x 的值为“tuvwxyz”，执行如下程序后，变量 res 的值不可能是：（ D ）

```
from random import randint
x="tuvwxyz"
res=""
i=0
j=len(x)//2
while j<len(x) and i<len(x)//2:
    if randint(0,1)==0:
        res+=x[i]
        i+=1
    else:
        res+=x[j]
        j=j+1
print(res)
```

A. tuvv B. twxyz C. wtuv D. twvxyz

x
↓
tuvwxyz
↑
y

程序功能：随机生成0或1，选择i位置或j位置的字符连入结果res
根据循环条件，i的范围到len(x)//2，D选项如果已连twuv，其中的tuv来自于i指向的字符，如果这三个字符都已连，循环条件就不成立了，就会退出循环了，因此不会有后面的字符

12. a=[[34, 1], [45, 3], [31, 0], [66, 5], [23, 2], [70, -1]]，该列表中的数据元素是一个升序链表，以下 python 代码段可以完成在该链表中插入新数据并保持升序特性，则划线处应填入代码为：（ C ）

```
a=[[34, 1], [45, 3], [31, 0], [66, 5], [23, 2], [70, -1]]
head=4
b=int(input("请输入需要插入的数据"))
p=q=head
```

a.append([b, -1]) **添加节点**

① **添加的节点在列表的最后，因此位置为len(a)-1**

程序功能

```
while ② :
```

```
    q=p
```

```
    p=a[p][1]
```

根据这两句代码，可知p为遍历链表的变量，q记录p前驱节点的位置
因此循环遍历的条件为p!=-1

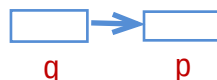
```
if p==head:
```

头插

```
    a[m][1]=head
```

```
    head=m
```

```
else:
```



```
    ③
```

```
    a[q][1]=m
```

中间插：
在q和p之间插入新节点
所以新节点的指针指向p,前面的q节点指向新节点所在的位置

```
print(a)
```

- A. ①m=len(a)-1 ②p!=-1 and a[p][0]<=b ③a[m][1]=q
B. ①m=len(a) ②q!=-1 and a[p][0]<b ③a[m][1]=p
C. ①m=len(a)-1 ②p!=-1 and a[p][0]<=b ③a[m][1]=p
D. ①m=len(a) ②p!=-1 and a[p][0]<b ③a[m-1][1]=p

二、非选择题（本大题共 3 小题，第 13 小题 7 分，第 14 小题 9 分，第 15 小题 10 分，共 26 分。）

13. 小李在学习了数的编码后，他用 python 编写了一个程序，可以实现十六进制转二进制运算，程序代码如下：

（1）将程序中加框处的代码，改为“`print(result)`”，则当输入十六进制值为“1A”时，输出为 [0,0,0,1,1,0,1,0]。

（2）请在划线处填入合适代码，完善功能。

```
def ltos(x):
```

函数功能：去掉转成二进制后，前面的0

```
    s=""
```

```
    for i in range(len(x)):
```

```
        s=s+str(x[i])
```

将列表中的数值逐一取出并连成字符串

```
    if s[0]=="0":
```

```
        for i in range(1,len(x)):
```

```
            if s[i]!="0":
```

如果第一位为0，遍历余下的位
如果当前位不为0，说明此位之前的均为0，
则取从当前位开始之后的所有字符
建议可只写开始位置，不用管末尾的位置

```
                ① s=s[i:]
```

```
            break    s=s[i:len(s)]
```

```
    return s
```

```
num=[8,4,2,1]
```

```
H=input("请输入一个十六进制值：")
```

程序功能：利用4位二进制各位上的权值
凑出一位十六进制的二进制表示形式

```
b=[0,0,0,0]
```

```
result=[]
```

```
for i in range(len(H)):
```

```
    if "0"<=H[i]<="9":
```

十六进制如果是10以内的，直接转为数值

```
        a=int(H[i])
```

```
    else:
```

② a=ord(H[i])-55 如果是A-F，利用ASCII码，将其转为对应的10-15

```
    for j in range(len(num)):    遍历4位权值8,4,2,1
```

```

    if a-num[j]>=0:
        b[j]=1
        a=a-num[j]
    result+=b
    ③ b=[0,0,0,0] 恢复初值

```

如果当前位上的值超过当前权值
则此位上写1
减掉当前权值，重复此过程

```
print("该十六进制的二进制值为：",ltos(result))
```

14.某公司正在开发一款新游戏，为了能够更好的开发，公司对其员工进行了一次问卷调查。调查的结果保存在 search.xlsx，部分内容如图 14-1。请你帮助该公司编写 Python 程序来进行统计。

姓名	职业	工龄	擅长平台	工作软件	擅长类型
程*	CG设计师	102	Web	Maya	RTS
陈*	游戏设计	10	Switch	虚幻引擎	FPS
懂**	程序设计	4	Windows	虚幻引擎	TAB
何*	程序设计	30	Windows	Unity	SOC
林**	美术设计	15	Android	SAI	RPG
李*	程序设计	7	Windows	Unity	SOC
宁**	游戏设计	14	Windows	Unity	FPS
彭**	游戏设计	11	IOS	虚幻引擎	RPG
任*	音频	8	Windows	SteelSeries	SOC
沈**	编剧	9	Windows	Scrivener	AAVG
申**	美术设计	5	Android	SAI	FPS
王**	程序设计	7	Linux	Unity	TAB
杨*	算法工程师	13	Windows	Visual Studio	FPS

图 14-1

(1) 检查发现第一条数据的“工龄”项存在逻辑错误，下列做法正确的是 B (单选)。

- 统计剩余的 CG 设计师工龄平均值，用平均值代替该数据。
- 向该 CG 设计师本人再次询问，确认其真实工龄，然后修改数据。
- 工龄数据可能已经遭到破坏，重新对所有人再次进行问卷调查，防止统计出错。

(2) 解决了数据的错误后，现在按照职业进行分类，要求统计各个职业的平均工龄和最常用的工作软件，统计结果列表 age 和 software 中。实现该功能的 Python 程序如下。

```

import pandas as pd
import matplotlib.pyplot as plt
career = ["游戏设计", "程序设计", "CG 设计师", "算法工程师", "音频", "美术设计", "
编剧"]
df = pd.read_excel("search.xlsx")
age, software = [0] * len(career), ["" ] * len(career)
k = 0
for crr in career:
    df1 = df[df["职业"] == crr] #在 df 中找到所有与 crr 匹配的数据
    df2 = df1.groupby("工作软件",as_index=False). 姓名.count()
    age[k] = df1["工龄"].mean() #找到出现次数最多的工作软件名称
    software[k] = 
    k += 1

```

注意这里是对按职业选出的数据求工龄平均，因此对象为df1
注意求平均的函数为mean()，括号不能省

```
for i in range(k):
    print("职业:", career[i], "平均年龄:", age[i], "最常用工作软件:", software[i])
```

请在划线处①②填入合适的代码。

为③处选择合适代码 B (单选)。

根据题目要求，找到出现次数最多的工作软件名称
因此用sort_values函数排降序，再找第一行的数据

A. df2.max()

B. df2.sort_values("姓名", ascending=False).工作软件[0]

C. df2.工作软件.max()

(3)在以上统计的基础上，该公司要确定最适合该公司开发的游戏类型。编写程序完成统计，并画出如图 14-2 的柱状图。实现该功能的代码如下，请在划线处填入合适的代码。

```
plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
```

```
plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
```

```
df2 = df.groupby("擅长类型").count() / len(df) * 100
```

```
df2.rename(columns = {"姓名": "人数"}, inplace=True)
```

```
plt.bar(df2.index, df2.人数, label = "游戏类型%")
```

```
plt.title("**公司游戏开发类型汇总")
```

```
plt.legend()
```

```
plt.show()
```

注意：这里groupby函数没有as_index参数，说明默认as_index=True，所以“擅长类型”列已作为索引，下面引用这列数据就只能用index，而不是列名称来引用了，一定要注意分组里的参数

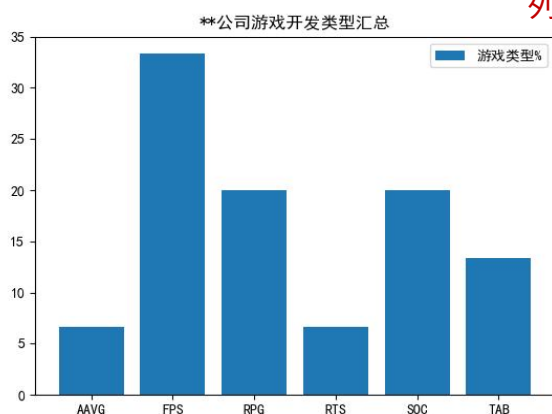


图 14-2

15. 某电子厂为智能手机组装供应屏幕组件，每条生产线一次会生产n块同型号屏幕，根据行业标准，每块屏幕需检测初始亮度值，亮度范围为0到100尼特(屏幕亮度单位)。根据最新质量规范要求：为保证同批次手机亮度体验的一致性，要求同一条生产线产出的所有屏幕，其初始亮度最大值与最小值的差不超过m尼特，否则品牌方拒绝验收本批产品。

屏幕初始亮度可通过专业设备进行校准调整，调整规则为：每调高1尼特初始亮度，需消耗成本v1元，每降低1尼特初始亮度，则需消耗成本v2元。现给定同批次n块屏幕的初始亮度（整数类型），存入列表a中。为顺利通过验收，编写程序计算该电子厂针对这批屏幕最少需要投入多少校准成本，并输出按该方案校准后的最低亮度和最高亮度，若有多个符合条件的最少成本方案，则选用最低亮度较小的方案。请回答下列问题：

(1) 若一批产品的初始亮度值a=[74, 69, 34, 64, 32]，控制m=10尼特，假设每调高1尼特需要消耗成本2元，每降低1尼特需要消耗成本1元。为通过验收，将校准后的最低亮度调整为34尼特，则至少需要投入 79 元成本。

根据最低亮度34,最大亮度值与最小亮度值差不超过10尼特,可知最大亮度为44,可知各产品需调的值为:

降30 降25 不变 降20 升2
总成本: 30+25+20+2*2=79

(2) 定义pre_process(a)函数，该函数功能为：统计原数据中，各个初始亮度的屏幕数量。找到这一批产品中的亮度最大值和最小值。

```
def pre_process(a):
    t=[0]*101
    mmax,mmin =0,101
    for i in range(len(a)):
        t[a[i]]+=1
        if a[i]<mmin:
            mmin=a[i]
        elif a[i]>mmax:
            mmax=a[i]
    return t,mmax,mmin
```

两个分支只选其一，可能导致初始最大值取不到
比如：74,69,34,64,32
程序执行时先判断a[i]<nmin，此时nmin为101，74小于101，选择if分支，结束，继续遍历下面的数据，这样下面找的最大值就排除了74，程序出错
用两个if语句就可以做两次判断，得到正确的结果

方框中语句有误，可修改为以下选项中的 A （单选，填字母）

A. if a[i]>mmax B. else C. elif a[i]>=mmax

(3) 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 cal(x,v1,v2,m): #计算最低亮度为x的校准成本
```

```
    s=0
```

```
    for j in range(mmin,x):
```

```
        s+=v1*t[j]*(x-j)
```

```
    for j in range(x+m+1,mmax+1):
```

```
        s+= ① v2*t[j]*(j-x-m)
```

```
    return s
```

''' 读取同批次n块屏幕的初始亮度，依次存入列表a中，代码略

读取m值，限定一个批次内亮度最大与最小的差值不超过m(允许等于m)，代码略

读取v1、v2，分别代表每调高、降低1尼特的成本价格，代码略

'''

```
t,mmax,mmin=pre_process(a) 找出各个初始亮度的屏幕数量，最大亮度，最小亮度
```

```
ans=max(v1,v2)*len(a)*100 #初始化为一个不可能的最大值
```

```
ansi=0
```

```
if mmax-mmin<=m:
```

```
    print("本批次产品无需校准，可直接通过验收")
```

```
else:
```

```
    for i in range(mmin,mmax-m+1): #枚举最低亮度
```

```
        ② s=cal(i,v1,v2,m)
```

```
        if s<ans:
```

```
            ans=s
```

找最小成本

```
    ansi=i
```

```
    print("通过验收的最低成本为：" + str(ans))
```

```
    print("该方案中，本批次产品的最低亮度为：" + str(ansi))
```

```
    print("该方案中，本批次产品的最高亮度为：" + ③ str(ansi+m))
```

根据最低亮度ansi和差值m，得到最高亮度