

1~ 5 CDABC

6~10 ADCBB

11~12 AB

高三下信息选考练习卷 6

一、选择题（本大题共 12 小题，每小题 2 分，共 24 分。每小题只有一个正确答案。）

阅读下列材料，回答第 1 至 4 题：

某博物馆搭建非遗数字平台，采用文本、图片、视频等多种媒体形式对传统技艺进行数字化存储，公众能够借助网站或 VR 眼镜深度体验。

1. 下列关于该平台中数据与信息的描述，正确的是 **C**
 - A. 图片能够毫无失真地还原实物外观细节 **数字化一定会失真**
 - B. 使用不同设备浏览同一传统技艺，获得的体验相同
 - C. 平台中的数据被拷贝后，其原始数据不会损耗
 - D. 非遗信息的价值对所有使用者都是相同的
2. 关于信息安全与信息社会责任，下列行为合适的是 **D**
 - A. 某公司未经授权，将平台图片用于商业广告设计
 - B. 为提高访问速度，可临时关闭防火墙
 - C. 在收集用户信息时，不告知其具体用途
 - D. 对重要数据加密存储，并定期备份
3. 该平台推出“AI 剪纸”功能，应用人工智能模型将用户上传的图片转换为剪纸风格的艺术图像，以下做法无助于提升艺术图像生成效果的是 **A**
 - A. 提升交互界面的用户友好性 **和图像生成效果无关，可以提高用户体验感**
 - B. 对图片进行裁剪以突出主体
 - C. 改进模型中提取图像轮廓的算法
 - D. 适当增加模型训练时的数据量
4. 馆内某展厅有 12 件陶瓷类、15 件编织类、25 件剪纸类和 9 件雕刻类展品。若使用二进制对这些展品进行编码，二进制的前几位表示类别，其余位表示该类别的具体展品，则所需的二进制位数最少是 **B**

3 个类别 2 位
具体展品最大的是 25, 5 位

A. 6	B. 7	C. 8	D. 17
------	------	------	-------

阅读材料，回答第 5 至 7 题：

某河道水质监测系统监测点集成了水质传感器、北斗定位模块、太阳能供电模块和物联通信模块，通过智能终端定时收集水质与位置数据后传输至服务器。当服务器判定某监测点水质数据超标时，会向管理员的手机 APP 发送示警信息。市民可通过 APP 查看各河段水质信息。

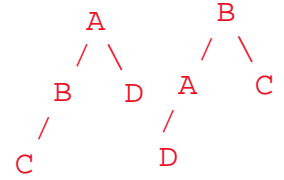
5. 下列关于该信息系统功能与应用的说法，正确的是 **C**
 - A. 该系统不会有安全隐患
 - B. 传感器获取水质数据属于系统的输出功能
 - C. APP 的使用便于管理员了解水质信息
 - D. 太阳能供电模块主要用于提升数据处理速度
6. 下列关于该信息系统中软件与网络的说法，正确的是 **A**
 - A. 安装了 APP 的移动终端一定有系统软件
 - B. 智能终端收集数据无需任何软件支持
 - C. 监测点与服务器之间的通信只能使用无线连接
 - D. 各河段水质信息不属于该系统的网络资源
7. 下列关于该信息系统中数据的说法，不正确的是 **D**
 - A. 示警信息的生成依赖于服务器对数据进行分析与判断
 - B. 系统中存储的历史水质数据可用于长期水质监测
 - C. 北斗定位模块损坏会影响数据的采集
 - D. 系统中数据的表现形式必须是相同的

8. 某栈 S 为空，队列 Q 从队首到队尾的元素依次为 1、2、3。约定有两种操作：操作 A 为队首元素出队并入栈，操作 B 为栈顶元素出栈并入队。经过多次操作后，若栈 S 从栈底到栈顶的元素依次为 1、3、2，则操作次数最少为 **C**

- A. 3 B. 4 C. 5 D. 6

9. 某完全二叉树有 4 个节点，分别标记为 ABCD，已知 A 是 D 的父节点，B 是 C 的父节点，则该二叉树的前序遍历序列可能为 **B**

- A. BCAD B. BADC C. ACBD D. ABDC



10. 有如下 Python 程序段：

```
def f(n, sign):
    if n == 0:
        return n
    return n * sign + f(n - 1, -sign)
```

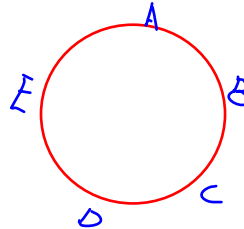
$f(5, 1) = 5 * 1 + 4 * (-1) + 3 * 1 + 2 * (-1) + 1 * 1 + 0 * (-1) = 3$

执行语句 print(f(5, 1))，输出的结果为 **B**

- A. -5 B. 3 C. 7 D. 15

11. 有如下 Python 程序段：

```
from random import randint
n = len(s)
res = ""
i, j = 0, n - 1
while len(res) < n:
    k = randint(1, n)
    if k % 2 == 0:
        res += s[i]
        i = (i + k) % n
    else:
        res += s[j]
        j = (j - k) % n
```



下一次如果 i 走动，和上一次 i 间隔一定是偶数

下一次如果 j 走动，和上一次 j 间隔一定是奇数

若字符串 s 的值为“ABCDE”，执行该程序段后，变量 res 的值不可能是 **A**
 AB间隔是奇数，不能是i连走2次，但是B不可能是第一次的j

- A. “ABCDE” B. “AECDB” C. “EADCB” D. “EDCBA”

12. 使用列表 d 模拟链表结构（节点数大于 2），每个节点包含数据区域和指针区域，h 为头指针。现要修改该链表各节点的链接关系，将链表 L0→L1→L2→⋯→Ln-1 重排为 L0→Ln-1→L1→Ln-2→⋯，实现该功能的程序段如下，运行结果如第 12 题图所示。

初始化链表数据 d，并输出链表数据，代码略

```
s = h    #h 为头指针
```

```
[ ]
```

```
print("重排后链表数值顺序:", end=" ")
```

```
p = h
```

```
while d[p][1] != -1:
```

```
    print(d[p][0], end="→")
```

```
    p = d[p][1]
```

```
print(d[p][0])
```

有如下代码段，在方框中填入的语句顺序应为 **B**

```
重排前链表数值顺序: 10→11→12→13→14→15
重排后链表数值顺序: 10→15→11→14→12→13
```

第 12 题图

1、先将链表用快慢指针分成两段：

10->11->12

13->14->15

2、再将第二段逆向

15->14->13 p2

3、链表合并：将 p2 中的节点逐个插入到 p1 中

p1: 10->11->12

p2: 15->14->13

-1 p cur->nxt p cur->nxt 10->11->12->13->14->15 p s f d[f][1]

先排除
③pq未定义

A. ③①②

B. ②①③

C. ①③②

D. ①②③

代码段①	代码段②	代码段③
<p>链表反向</p> <pre> p = -1 cur = s while cur != -1: nxt = d[cur][1] d[cur][1] = p p = cur cur = nxt </pre>	<p>f快指针, s慢指针 找中间位置s/p</p> <pre> f = h while f != -1 and d[f][1] != -1: p = s s = d[s][1] f = d[f][1] if f != -1: 无用语句 f = d[f][1] d[p][1] = -1 </pre> <p>f → s</p>	<p>链表合并</p> <pre> p1 = h p2 = q = p while q != -1: q = d[p2][1] d[p2][1] = d[p1][1] d[p1][1] = p2 p1 = d[p2][1] p2 = q </pre> <p>将p2插入到p1的链中</p>

二、综合题 (本大题共 3 题, 第 13 题 9 分, 第 14 题 8 分, 第 15 题 9 分, 共 26 分)

13. 某停车场管理系统, 在停车场内安装有多个智能终端, 每个停车位上安装一个传感器, 每个智能终端连接多个传感器。智能终端将传感器采集的数据通过网络上传至服务器。服务器处理数据, 并通过智能终端控制停车位指示灯亮灭, 以指示各停车位的实时占用情况。请回答下列问题:

- 可对 A 设置唯一编号以区分不同的停车位。(单选, 填字母: A. 传感器 / B. 智能终端)
- 下列能 有效 降低服务器处理数据压力的措施是 B。(单选, 填字母: A. 减少传感器采集间隔时长 / B. 仅当车位状态改变时才上传数据 / C. 将数据进行压缩后传输)
- 下列关于该系统设计的说法, 正确的有 AD。(多选, 填字母)。(注: 全部选对的得 2 分, 选对但不全的得 1 分, 不选或有选错的得 0 分)

- 系统设计时需考虑软件的可扩展性
- 该系统若增加一个车位, 必须增加一个智能终端
- 智能终端和服务器不能在同一局域网内
- 该系统可使用 B/S 架构实现

(4) 停车位上安装的传感器是为了识别停车位上是否有车辆停放, 请说出一种可用的传感器名称, 并说明它是如何识别的。

摄像头: 通过拍照分析车位图像, 根据车辆识别来判断是否停放; **距离传感器:** 安装在地面 (或顶部), 通过计算与前方的距离来判断车辆是否停放; **压力 (重量) 传感器:** 通过感知轮胎压力引起的形变或重量变化判断车辆是否停放 **地感线圈**

(5) 停车费用计算规则为: 首小时 5 元, 之后每小时 3 元, 不足 1 小时按 1 小时计, 单日 (24 小时) 最高收费 35 元。车辆连续停放超过 24 小时的视为第二次停放, 重新计费。例如: 停车 25 小时, 费用为第一天 35 元 (24 小时)+ 第二天 5 元 (1 小时), 共 40 元。

计算停车费用 fee 的部分 Python 程序如下。

获取并计算某车的停车小时数, 保存在 h (h 为整数且 h>0) 中, 代码略

d = h // 24

t = h % 24

fee = 0

if t == 1:

fee = 5

elif t > 1: 5+(t-1)*3 或 3*t+2 (25-int(26-t))*3+t 5+(ceil(t)-1)*3

fee = ▲

5+t//1*3? 要取整, 用int()

if fee > 35:

fee = 35

fee += d * 35

如果t为0时, fee=0被删除, 就没有fee的赋值语句, 报错

①请在程序中划线处填入合适的代码。

24 或 24的整数倍

②若程序中删除加框处语句, 则当 h 的值为 ▲ 时, 程序会报错。(列举一种情况)

14. 某用电监测系统采集了部分房间 2025 年的照明用电数据并存储于 data.xlsx 文件中, 如图 a 所示。现要找出月均照明用电量最多的房间 (各房间用电总量均不相同), 并绘制该房间 2025 年各月份的照明用电量折线图 (如图 b 所示), 并按用电量降序输出该房间各月份用电情况 (如图 c 所示)。应有groupby

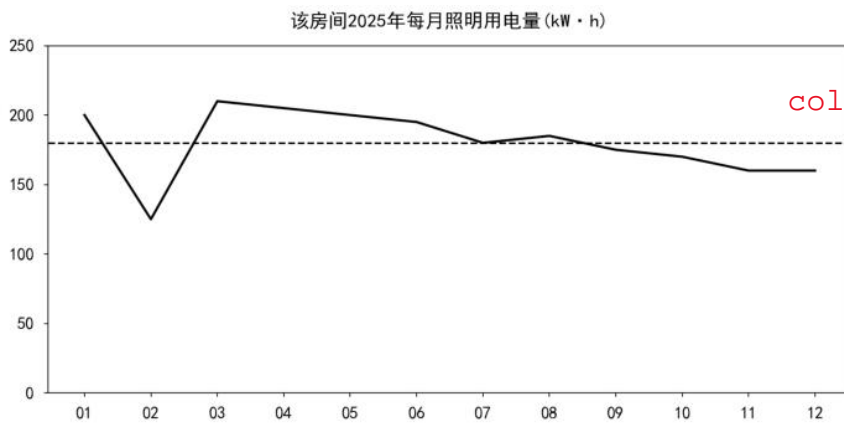
	A	B	C	D	E	F	G	H	I
1	时间	2-101	2-102	2-103	2-104	2-201	2-202	2-203	2-204
2	2025-01-01	1.99	4.96	5.96	1.35	5.19	5.56	3.24	3.45
3	2025-01-02	6.73	3.35	2.87	4.97	5.52	5.16	6.51	5.87
4	2025-01-03	0.74	4.89	5.58	5.68	7.21	6.39	6.53	6.67
5	2025-01-04	11.38	3.59	4.58	5.07	6.97	5.69	6.5	6.23
364	2025-12-29	7.36	5.08	3.65	4.49	5.5	6.32	0	6.46
365	2025-12-30	5.68	1.86	4.94	4.78	5.11	6.08	0	5.85
366	2025-12-31	4.08	4.52	5.02	4.29	5.6	3.37	0	3.09

Excel中 列号

	A	B	C	D	E	F	G	H	I
1	列1	列2	列3	列4	列5	列6	列7	列8	列9
2	刷	套	模	拟	卷	技	术	一	百
3	做	遍	真	题	卷	信	息	满	分
4	我	爱	乐	中	考	上	9	8	5

行号

第 14 题图 a



第 14 题图 b

导入df中时:

df.columns() 得到 列号Series

columns中存["2-101", "2-102",]

实现上述功能的部分 Python 程序如下。

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("data.xlsx") #读取文件
columns = df.columns[1:] #取得所有房间列表
maxv = 0 #columns中存["2-101", "2-102", .....]
for col in columns:
    s = df[col].sum() #求总和 求的每个房间号全年总用电量
    ave = round(s/12, 2) #保留 2 位小数 总量/12个月=月均用电量
    if ave > maxv:
        ① maxv = ave
        maxcol = col #记录月均最大用电量的房间号及月均值
print("月均照明用电量最多的房间为", maxcol)
df.insert(0, "月", "") #在首列插入"月"列
for i in df.index:
    t = df.at[i, "时间"]
    df.at[i, "月"] = ② t[5 : 7] #通过行标签和列标签选取单个值
df = df.drop("时间", axis=1) #删除"时间"列
df2 = df[["月", maxcol]] #从 df 中筛选出两列数据
```

月均照明用电量最多的房间为 2-201
该房间 2025 年每月用电情况如下:
月份 月用电量(kW·h)
03 210.6
04 210.24
05 203.85
06 200.11
01 196.25
08 186.07
07 179.95
10 172.71
09 172.17
12 156.67
11 155.63
02 130.36

降序

月	时间	2-101	2-102	2-103	2-104	2-201	2-202	2-203	2-204
/	2025-01-01	1.99	4.96	5.96	1.35	5.19	5.56	3.24	3.45
/	2025-01-02	6.73	3.35	2.87	4.97	5.52	5.16	6.51	5.87
/	2025-01-03	0.74	4.89	5.58	5.68	7.21	6.39	6.53	6.67
/	2025-01-04	11.38	3.59	4.58	5.07	6.97	5.69	6.5	6.23
/	2025-12-29	7.36	5.08	3.65	4.49	5.5	6.32	0	6.46
/	2025-12-30	5.68	1.86	4.94	4.78	5.11	6.08	0	5.85
/	2025-12-31	4.08	4.52	5.02	4.29	5.6	3.37	0	3.09

当前df2中只有index、月、maxcol三列
(其他房间号的列都删除了, 只留下月均最大的房间号)

① B. df3 = df2.groupby ("月", as_index = False).sum()
按月分类求和

#设置绘图参数, 代码略

② `F.plt.plot (df3.月,df3 [maxcol])` 提取两列做图

`plt.show()`

`print("该房间 2025 年每月用电情况如下:")`

`print("月份", "月用电量(kW·h)")`

③ `C.df3 = df3.sort_values (maxcol , ascending=False)` 根据图c降序排

`for i in df3.index:`

`print(df3.at[i, "月"], "", round(df3[maxcol][i], 2))`

(2) 请选择正确的代码填入方框处 (单选, 填字母)。

程序方框中①②③处可选的代码有:

A. `df3 = df.groupby ("月", as_index = True).mean ()` #分组求平均

✓ B. `df3 = df2.groupby ("月", as_index = False).sum ()` #分组求和

C. `df3 = df3.sort_values (maxcol, ascending=False)` #降序排序

D. `df3 = df3.sort_values (col, ascending=False)` `col`值为最后1列标题"2-204"

E. `plt.plot (df3.index, df3 [maxcol])` #绘制折线图

F. `plt.plot (df3.月, df3 [maxcol])`

(3) 2-201 房间 2025 年月均用电量为 180.67 kW·h, 则该房间超过月均用电量的月份有 6 个。

15. 某物流仓库提供货物搬运功能, 每件货物规格相同, 仓库内共有 n 辆叉车 (编号为 $1 \sim n$)。现收到若干个订单, 每个订单包括装货时间 (从 0 开始编号) 和货物数量。根据叉车调度规则, 要求在截止时间 t 前完成所有订单的货物搬运, 计算需要的最少叉车数量。**装货: 1件耗时1单位, 运送+卸货: 1批耗时3单位**

不能混装! 已知每辆叉车一次处理一个订单的货物, 且一次最多装载 10 件货物。叉车获取每件货物需花费 1 个单位时间, 每次将叉车上的所有货物搬运至货车中需花费 3 个单位时间。叉车调度规则如下:

①选择最早空闲叉车; 若空闲叉车有多辆, 则选择编号最小的叉车。

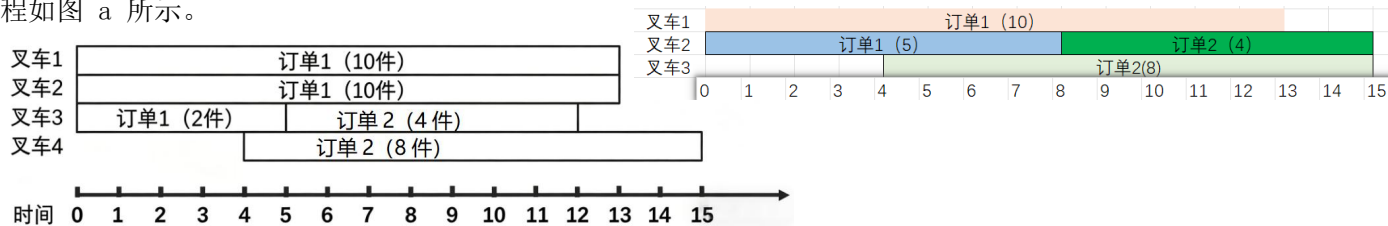
**无法完成, 本车必须尽可能装
剩余也不能混装**

贪心算法

②若一辆叉车无法在截止时间 t 前完成搬运, 则在每辆叉车尽可能多装货物 (一次不超过 10 件)

的前提下, 安排最少数量的叉车在截止时间 t 前完成; 若不能, 则输出 "订单无法完成!"。

例如 t 为 15, 订单序列 `data` 为 `[[0, 22], [4, 12]]`。当叉车数量为 4 时, 根据调度规则, 货物搬运过程如图 a 所示。



请回答下列问题:

(1) 若 `data` 为 `[[0, 15], [4, 12]]`, t 为 15, 则完成所有订单至少需要 3 辆叉车。

(2) 定义如下 `psort (data)` 函数, `data` 列表的每个元素包含 2 个数据项, 依次为装货时间和货物数量, 已按装货时间升序排列。函数的功能为保持订单按装货时间升序排列不变, 当装货时间相同时, 按货物数量降序排列, 返回 `data`。**按时间区段已有序, 有序区间内按关键字2 (货物数量) 降序排序**

`def psort (data):`

`for i in range (len (data)-1):`

`for j in range (len (data)-i-1):`

`if data [j][0] == data [j+1][0] and data [j][1] < data [j+1][1]:`
`data [j], data [j+1] = data [j+1], data [j]`

`return data`

若关键字1 时间, 关键字2都无序下

按时间区升序序, 时间相同则按关键字2 (货物数量) 降序排序

`data [j][0] > data [j+1][0] or data [j][0] == data [j+1][0] and data [j][1] < data [j+1][1]`

要实现函数功能，方框处应填入的代码为 C (单选，填字母)。

- A. data [j][1]<data [j+1][1]
- B. data [j][1]> data [j+1][1]
- C. data [j][0] ==data [j+1][0] and data [j][1]<data [j+1][1]
- D. data [j][0] != data [j+1][0] and data [j][1]>data [j+1][1]

(3) 实现计算最少叉车数量的部分 Python 程序如下，请在划线处填入合适的代码。

```
def findi(cars):
    mini = 0
    for j in range(len(cars)):
        if cars[j] < cars[mini]:
            mini = j
    return mini
```

①选择最早空闲叉车；若空闲叉车有多辆，则选择编号最小的叉车。

def check(data, m, v, t): m是叉车数量, v是最大运货量, t是截止时间

```
cars = [0] * m
```

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

```
    ① count=data[i][1] 不能做累加
```

```
    while count > 0: 如果本任务能运送, 最后count为0
```

```
        mini = findi(cars) 确定真正装车时间
```

```
        stime = max(cars[mini], data[i][0]) #max(a,b)返回 a,b 的最大值
```

```
        remain = t - 3 - stime 截至时间-卸货时间(3)-max(叉车最早空闲时间,货物到达时间)
```

```
        if remain <= 0: 任务 无法完成
```

```
            return False 同类型值比较, 说明count和v一样, 是关于运货量
```

```
            load = min(count, remain, v) #min(a,b,c)返回 a,b,c 的最小值
```

```
            ② car[mini]=stime+load+3 如果可以装车, 就更新装车的叉车时间
```

```
            count -= load 注意: 装车的开始时间不一定是car[mini]
```

```
    return True 错误: car[mini]=car[mini]+load+3
```

"""读取截止时间 t、叉车数量 n；读取订单数据表存入 data 列表，每个元素包含 2 个数据项，依次为装货时间和货物数量。代码略。"""

```
data = psort(data)
```

```
v = 10 叉车最大运货量
```

```
left, right = 1, n #叉车最大装货量
```

```
while left <= right:
```

```
    m = (left + right) // 2
```

```
    if check(data, m, v, t): right指向不可能的叉车用量
```

```
        right = m - 1
```

```
    else:
```

```
        left = m + 1
```

```
    if ③ left>n ! left==n+1 错误:left>right
```

```
        print("订单无法完成!")
```

```
    else:
```

```
        print("最少需要", left, "辆叉车")
```

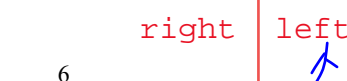
用对分查找找最少叉车数量

如果m量叉车无法满足, 会找到最后

错误:left>right



如果找到中间停下来, 此时left>right, m在停止线左边还是右边不确定



1. 贪心算法 常见不考虑装货时间

但是不是全局最优解

(例如可能最后某个订单, 1辆车在装卸, 其他叉车全闲置)

本题变化:

加入搬运特殊条件:

装货要1件1件装, 耗时1单位

卸货: 整批卸货, 耗时3单位

思考2: 能否不用对分, 直接遍历任务, 不满足叉车+1 ?

也可以写成:
t-stime<4