绝密★考试结束前

技术（二）

本试卷分两部分，第一部分信息技术，第二部分通用技术。满分 100 分，考试时间 90 分钟。

第一部分　信息技术（共 50 分）

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）

阅读下列材料，回答第 1 至 3 题：

2025 年 4 月 19 日，全球首场“人机共跑”半程马拉松赛事正式开跑，让全世界为之瞩目和惊叹。参赛机器人需要通过摄像头实时识别路标（如箭头、数字等）来决定行进方向，所以在前期进行了大量的数据训练。

1.下列关于该活动中数据和信息的说法，正确的是　D

A.“2025 年 4 月 19 日”在以上材料中没有意义

B.机器人训练的所有数据都具有同等的价值

C.机器人在比赛的过程中不会产生新的数据

D.路标信息必须依附于箭头、数字等载体存在

2.下列关于提高机器人路标识别准确率的方式，正确的有　A

A.增加训练数据的多样性和数量　　B.更换更轻便的机器人外壳材料

C.提高机器人的运行速度　　D.定期清理机器人内存中的缓存文件

3.下列关于人工智能的说法，正确的是　B

A.在符号主义人工智能框架下，实现摄像头路标的识别

B.在行为主义人工智能框架下，可通过环境反馈实时调整运动轨迹

C.该机器人参赛模型可迁移至多个领域，体现了混合增强人工智能应用　跨领域人工智能

D.参赛机器人完成马拉松赛事说明人工智能未来将完全取代人类

阅读下列材料，回答第 4 至 6 题：

某中学开发了“智慧校园停车管理系统”。该系统通过电子显示屏和手机 APP“慧校园”实时显示空余车位信息，根据车载电子标签自动识别车辆信息，引导车辆快速停放。系统根据服务器数据，可自动生成每日车流量报表。

4.下列关于信息系统的组成，说法正确的是　B

A.“慧校园”属于系统软件

B.摄像头和电子显示屏都属于该信息系统的硬件

C.车位信息存储于电子显示屏和手机 APP 中　信息系统中长期保存的数据都应存储于数据库中

D.电子显示屏的信息更新不需要网络支持

5.下列关于信息系统的功能，说法不正确的是　B

A.通过 RFID 技术获取电子标签中的车辆信息，体现了数据输入功能

B.为提高车牌识别速率，系统可采用红外通信技术进行车牌信息验证　提高识别速度可通过提高网速、识别算法等

C.系统自动生成每日车流量报表，体现了数据加工处理功能

D.服务器的 cpu 性能会影响车牌识别的速度

6.下列关于该信息系统安全的说法，正确的是 C

A.该系统采集的车辆行驶轨迹信息不属于个人敏感数据

B.车辆的通行记录应允许所有教职工查询

C.定期备份数据有助于保障信息系统中存储数据介质的安全性

D.为加快系统运行速度，应关闭防火墙

7. 某快递保管规则如下：物品存放的前 20 小时（包含）免费；超过 20 小时但未满 48 小时（包含）的部分收取 5 元基础费；若存放时间超过 48 小时，则超出部分按每 12 小时 3 元的标准累加计费（不足 12 小时按 12 小时计算）(math.ceil()函数为向下取整）。以下不符合上述计费规则的算法代码为 B

A.
```
if  T <= 20:
    m=0
elif  T <= 48:
    m=5
else:
    m=5 + 3 * math.ceil((T-48)/12)
```

B.
```
if  T <= 20:
    m=0
if  20< T <= 48:
    m=5
else:
    m=5 + 3 * math.ceil((T-48)/12)
```
T<=20 and T>48

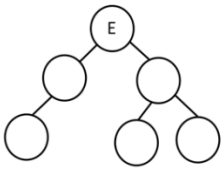
C.
```
if  T <= 20:
    m=0
else:
    if  T <= 48:
        m=5
    else:
        m=5 + 3 * math.ceil((T-48)/12)
```

D.
```
m=0
if  20<T <= 48:
    m=5
elif  T>48:
    m=5 + 3 * math.ceil((T-48)/12)
```

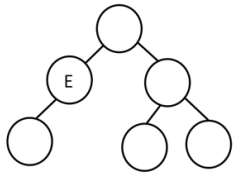
8.某队列操作规则如下：T 将队首元素出队；Q 将队首元素出队再入队。若经过 TQQTTQQTTQ 操作后，队列还剩 5 个元素，则该队列原始共有几个元素 D 出队5次，剩5个，原来10个

A.7　　B.8　　C.9　　D.10

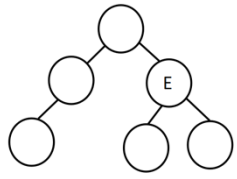
9.将下列二叉树按照前序遍历的规则压入栈中，则下列选项中，节点 E 在第二个出栈的为 D



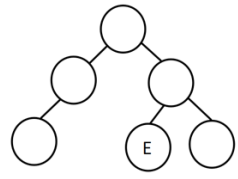
A.



B.



C.



D.

前序遍历E在倒数第二位

10.某 Python 代码段如下：

```
def f(x, n):
    if n == 1:
        return 1
    elif  n<4:
        return f(x, n//2)
    else:
        return f(x, n//2) * x
print(f(2, 8))
```

f(2,8)=f(2,4)*2=f(2,2)*2*2=f(2,1)*4=1*4

运行该程序段后，输出的值为 B

A.2　　B.4　　C.6　　D.8

11.有如下 Python 程序段：

```
import random
a=[34, 12, 78, 45, 23, 56, 89, 11, 67, 90]
n=len(a)
x = random.randint(1,5)*2
for i in range(n-1):
    for j in range(n-i-1):
        if  j < x-1 and a[j] > a[j+1]:
            a[j], a[j+1] = a[j+1], a[j]
        elif  j >= x and a[j] < a[j+1]:
            a[j], a[j+1] = a[j+1], a[j]
print("排序后数组:", a)
```

[2,10]之间的偶数

在0~x-1区间内升序排，x~n-1区间降序排

运行该程序段后，列表 a 的值不可能为

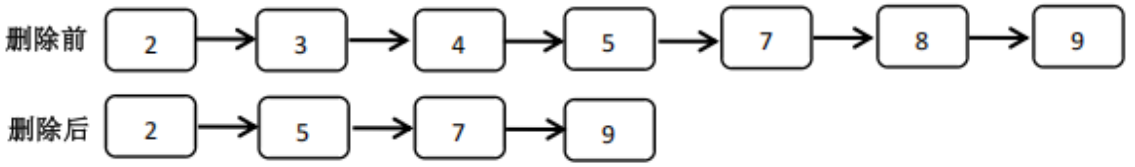
A.[12, 34, 90, 89, 78, 67, 56, 45, 23, 11]　x=2

B.[12, 34, 45, 78, 90, 89, 67, 56, 23, 11]　x=4

C.[11, 12, 23, 34, 45, 56, 78, 89, 90, 67]　x=8

D.[12, 34, 78, 90, 89, 67, 56, 45, 23, 11]　x=3

12.利用列表 a 来模拟链表结构（节点数大于 0），每个节点包含数据区域和指针区域，head 为头指针，各节点已按数据区域中的数值由小到大排列，现要对该链表中每段连续递增的中间节点数据进行删除，如下图所示，实现该功能的程序段如下，则方框中应填入的正确代码为：



第 12 题图

```
data = [[3,3],[2,0],[7,5],[4,4],[5,2],[8,6],[9,-1]]
head = 1
t = head
while t != -1:
    p=data[t][1]
    if data[t][0] + 1 == data[p][0]:
        next_ p = data[p][1]
        while next_ p != -1 and data[next_ p][0] == data[p][0] + 1:
            [          ]
        data[t][1] = p
    t = data[t][1]
```

t　p　next_p 三者顺序

p next_p

p　next_p

t是连续区域的头，p是连续区域的尾，中间都跳过

A.	B.	C.	D.
p = next_ p next_ p = data[p][1]	next_ p = data[p][1] p = next_ p	data[t][1]= data[p][1] p = next_ p	p = next_ p data[t][1]= data[p][1]

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

13. 某游戏玩家拥有 n 件装备（编号 0~n-1），每件装备提供攻击力和防御力，并占用 1 格背包空间及一定重量。给定背包容量 s（最大格数）和承重上限 w，装备数据以"编号 攻击力 防御力 重量"格式给出，例如"0 5 2 3"表示 0 号装备攻击力 5、防御力 2、重量 3。现玩家规则如下：

1.每件装备要么选择，要么不选；

2.占用背包总格子数 ≤s，且总负重 ≤w。

现要求设计程序，选择若干件装备，在满足总格数≤s 且总重量≤w 的前提下，使攻击力与防御力之和最大。

（1）若 n=3,s=2,w=6,编号 0~2 的装备重量依次为 3、1、5，攻击力依次为 5、4、2，防御力依次为 2、3、4 则选取编号__0，1_____装备。

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

```
#读取装备总数量，存入 n；读取背包总格子数，存入 s；读取总允许负重，存入 w，读取代码略
#读取各装备攻击力防御力及重量，存入列表 d，如 d=['0523','1431','2245','3367']，代码略
max_v=0
i=1
while i <2**n:
    wsum=0;vsum=0
    x=i;j=0;change=[]
    while x>0:
        r=x%2
        if r==1:
            wsum+=int(d[j][3])
            vsum+=int(d[j][1])+int(d[j][2])  ①
            change.append(j)
        x=x//2  ②
        j+=1
    if vsum>max_v and len(change)<=s and wsum<=w:
            ans=change  ③
            max_v=vsum
    i+=1
print("最高总价值为：",max_v)
print("选择商品编号为：",ans)
```

编号	重量和	攻击力与防御力之和
0, 1	4	9+5=14
0, 2	8	7+6=13
1, 2	6	6+7=13

i=1(1B),只要第一件物品
i=2(10B)只要第二件物品
i=2(11B)要一、二物品

#将该装备的防御力和攻击力加上，存储到vsum变量中

将所有可能的组合数一一列举，然后逐一去验证是否满足要求，而在列举的过程中根据i=1，while i <2**n:判断可知，是用二进制的方式来进行列举，共用 n件物品， 共有2**n 种不同的选择情况，当二进制的值为 1 时，代表选择，为 0 时代表不选择，将所有可能的情况一一列举，然后逐一验证，

14. 为了优化学校食堂用餐资源配置、改善用餐环境，学校科研小组搭建“智慧食堂”。该系统能够准确称量所选餐品的重量，获取餐品的种类，智能终端获取数据后，通过无线通信方式将数据传输到 web 服务器，请回答下列问题。

（1）在该信息系统搭建的前期准备中，以下关于该系统设计功能的说法，正确的是ADE(注:全部选对得 2 分，选对但不全得 1 分，不选或错选得 0 分)

A.获取餐品种类可用 RFID 技术实现　一般是不同种类的餐品盘子下的电子标签是不同的

B.获取餐品重量可用红外传感器实现　重力传感器

C.根据餐品重量和种类计算出餐品金额只能在服务器端完成　也可以在智能终端算

D.智能终端采用 WiFi 传输数据主要因其传输距离远、速率高

E.餐品金额和种类等数据可存储在数据库中

F.该系统网络应用软件的实现架构为 C/S　B/S

（2）若系统使用 GET 方式将采集的商品种类为'potato'，价格为 3 的数据上传服务器，使用的 URL 为 http://192.168.180.3:8080/s?zl=potato&price=3,服务器端相应的代码如下：

```
from flask import Flask
app = Flask(_ name_ )
```

①

```
def index(    ) :
    #获取上传的商品种类和价格数据，并进行处理，代码略”
if __name__=='__main__' :
```

②

若①、②处可填入以下代码:

A.@app. route("/")

B.app.run (host="192.168.180.3")

C.@app. route("/s")

D.app.run (host="192.168.180.3:8080")

E.@app. route(" /s?")

F.app.run (host="192.168.180.3",port=8080)

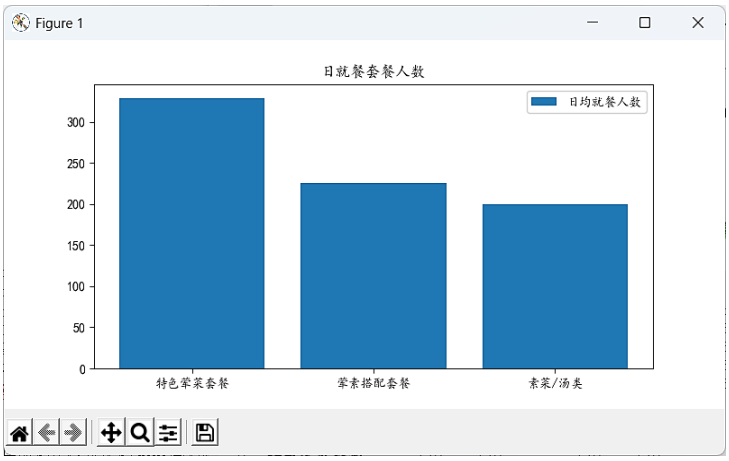
则①处代码为C(单选, 填字母)，②处代码为F(单选, 填字母)。

（3）系统搭建完成后，科研小组开始进行系统测试，发现 Web 服务器的响应较慢，在不考虑更换服务器的前提下，可以采取哪些措施来提升服务器的性能：_____。（注:回答 2 项，1 项正确得 1 分）　更换性能更好的cpu，扩大内存容量

（4）系统测试运行后，科创小组导出了各食堂 4 月 14 日数据，部分如图 a 所示，现要统计一楼食堂 14 日售卖数量最高的前 3 名套餐，并绘制柱形图如图 b 所示, 部分 python 程序如下:

	A	B	C	D	E	F	G	H
1	订单号	姓名	实付金额	门店	账户编码	支付日期	支付时刻	菜品
2	f0000000160426730	周泽镥	2	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:07	米饭/小菜
3	f0000000160426016	蒋天敏	1.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:06	米饭/小菜
4	f0000000160425636	潘闻涛	13.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:06	精品套餐
5	f0000000160425191	张今津	3.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:05	豪华套餐
6	f0000000160425163	吴逸轩	11.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:05	特色荤菜套餐
7	f0000000160424740	周泽镥	10.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:05	特色荤菜套餐
8	f0000000160422818	沈锦城	10.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:03	特色荤菜套餐
9	f0000000160422673	茹鑫倩	2.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:03	米饭/小菜
10	f0000000160422502	钟昊然	8.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:03	荤素搭配套餐
11	f0000000160422169	赵旖	6	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:03	素菜/汤类
12	f0000000160422073	朱俊晖	11	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:03	特色荤菜套餐
13	f0000000160421890	朱盼奕	0.5	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:03	米饭/小菜
14	f0000000160421563	欧志彤	12	一楼食堂	2.02E+13	2025-04-14 00:00:00	12:02	特色荤菜套餐

第 14（4）题图 a



第 14（4）题图 b

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_excel("202504.xlsx")  # 读取文件 202504.xlsx 中的数据
df=_____          #选出一楼食堂的数据
```

```
plt.bar( df2.菜品 , df2.订单号,label="日均就餐人数")
plt.title("日就餐套餐人数")
plt.legend()
plt.show()
```

①画线处合适的代码为C（单选,填字母）

A.df[df["门店"]]=="一楼食堂"

B.df["门店"=="一楼食堂"]

C.df[df["门店"]=="一楼食堂"]

D.df[df."门店"=="一楼食堂"]

②要实现上述功能，画框处合适的代码顺序为ACE（选 3 项, 填字母, 少选、多选、错选或次序错均不得分）

A.df1=df.groupby("菜品",as_index=False).count()

B.df1=df.groupby("菜品",as_index=False).sum()

C.df1=df1.sort_values("订单号",ascending=False)

D.df1=df1.sort_values("订单号",ascending=True)

E.df2=df1.head(3)

F.df2=df1.tail(3)

15. 某大学图书馆有 M 个自习室座位，这些座位可以划分为两个区域：安静区和讨论区。现在有 N 个学生申请使用座位，每个学生有开始使用时间 start 和结束使用时间 end（start< end），以及预约区域类型。管理员需要根据每日预约情况提前决定将多少座位分配给安静区（x 个），多少分配给讨论区（M-x 个），以最大化满足学生需求，并计算最多可满足的学生个数。

第 15 题图

如上图所示，若某天座位预定时间如下，（08:30:00 12:00:00 1）代表预约开始使用时间 08:30:00 点，结束使用时间 12:00:00 点，1 表示预约区域为安静区，0 表示预约区域为讨论区，若有多个空闲座位，优先选择座位编号较小的使用，现要根据如上预约计算最合理的座位分配，请回答下列问题：

（1）若可预约座位数为 3，则根据上图 8 条预约记录，最多可满足的申请数量为_7____。

（2）定义函数 readfile()读取 file 文件，该函数的功能是根据预约记录区分安静区和讨论区预约情况，请在划线处填入合适的代码

方案 1	时间段	方案 2	时间段
安静区 1	08:30:00 12:00:00	安静区 1	08:30:00 12:00:00
	13:45:00 16:15:00		13:45:00 16:15:00
安静区 2	10:00:00 13:30:00	讨论区 1	09:15:00 11:45:00
	15:45:00 18:15:00		12:30:00 15:00:00
讨论区 1	09:15:00 11:45:00		15:45:00 18:15:00
	12:30:00 15:00:00	讨论区 2	14:30:00 17:00:00
	16:30:00 19:00:00		

```
def readfile(file):
    file = open(file, 'r')
    quiet = []   # 安静区申请
    discuss = []   # 讨论区申请
    for line in file:
        parts = line.strip().split()#读取文件的一行去除换行符并按空格分隔
        start = int(parts[0][:2])*60+ int(parts[0][3:5])
        end = int(parts[1][:2])*60+int(parts[1][3:5])
        zone = int(parts[2])
        if zone == 1:
            quiet.append([start, end])
        else:
            discuss.append([start, end])
    file.close()
    return quiet,discuss
```

（3）定义函数 simple_sort(arr)，参数 arr 是安静区或讨论区的预约情况。

```
def simple_sort(arr):
    for i in range(len(arr)):
        min_idx = i
        for j in range(i+1, len(arr)):
            if arr[j][0] < arr[min_idx][0]:
                min_idx = j
        if min_idx!=i:
            arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

选择排序

一共4条数据只有第3和4顺序需要交换

若预约数据如第 15 题图所示，要对安静区数据进行排序，则加框处代码一共执行次数为_1____次。

（4）实现分配座位功能的部分 Python 程序如下，请在划线处填入合适的代码.

```
def al_seats(apps, count):                    统计apps中的预约
    seats = [-1] * count    # 初始化座位
    count_s = 0    # 成功分配的申请数
    for a in apps:   #遍历当前区域的预约数据
        start = a[0]   #start存储当前预约记录的起始时间
        end = a[1]    #end 存储当前预约记录的结束时间
        flag = False
        for i in range(count):    #遍历所有座位，检查是否有空闲座位可分配。
            if seats[i]==-1:    #若当前座位还未被分配过
                seats[i] = end     #当前座位更新为其结束时间
                count_s += 1     #可分配人数+1
                flag = True     #标记已分配
                break
        if not flag and count > 0:     #当没有空闲座位时则去寻找分配座位中最早结束的座位
            earliest = seats[0]
            idx = 0
            for i in range(1, count):
                if seats[i]<earliest:    #①
                    earliest = seats[i]
                    idx = i
            if earliest < start:   # 如果最早结束的座位的结束时间早于当前预约的开始时间，则替换该座位的预约
                count_s+=1
                seats[idx]=end    #②                #将当前座位时间替换成该预约的结束时间
    return    count_s
m=int(input("请输入图书馆座位数量："))
quiet,discuss=readfile('bookings.txt')
simple_sort(quiet)
simple_sort(discuss)
max_count = 0
for x in range(m + 1):    #变量 m 为座位总数量，x 变量座位总数量，从 0-m 每种情况判断
    q_count =al_seats(quiet, x)
    d_count =al_seats(discuss, m -x)    #③
    if q_count + d_count > max_count:
        max_count = q_count + d_count
        max_q=x
        max_d=m-x
print("最多可以满足的学生申请数量:", max_count)
print("此时安静区分配:",max_q,"个","讨论区分配:",max_d,"个")
```

#条件成立时更新时间，则应该是当前时间小于该遍历的座位结束时间，①空为 seats[i]<earliest,题干中说，若结束时间相同选座位编号较小的，故结束时间相等时不替换。

#al_seats 函数为根据输入的数据和座位返回可以满足学生的人数,则q_count变量存储该方案安静区满足的人数