

2021 年上海市高等学校信息技术水平考试试卷
一级（大学信息技术+人工智能基础）（试卷实例）

（本试卷考试时间 90 分钟）

一、单选题

（本大题 25 道小题，每小题 1 分，共 25 分），从下面题目给出的 A、B、C、D 四个可供选择的答案中选择一个正确答案。

1. 直接连接存储是常用的存储形式，主要存储部件有_____。

- A. 硬盘
- B. 移动硬盘
- C. 网盘
- D. U 盘

2. 十六进制数 DEFH 转换为十进制数是_____。

- A. 3566
- B. 3567
- C. 3568
- D. 3569

3. 信息安全是指计算机系统的硬件安全、软件安全和_____。

- A. 操作安全
- B. 运行安全
- C. 管理安全
- D. 数据安全

4. 辅助存储器包括_____等。

- A. 磁盘存储器、磁带存储器、光盘存储器
- B. 磁盘存储器、CMOS、Cache
- C. 磁盘存储器、BIOS、Cache
- D. U 盘和寄存器

5. 以及时响应外部事件为主要目标的操作系统是_____。

- A. 批处理操作系统
- B. 实时操作系统
- C. 分时操作系统
- D. 网络操作系统

6. _____不属于人工智能的重要应用。

- A. 博弈
- B. 专家系统
- C. 机器翻译
- D. 无线同屏

7. _____不属于外部设备。

- A. 光盘
- B. 硬盘
- C. 声卡
- D. 内存

8. _____不属于按网络的覆盖范围进行分类。

- A. 公用网
- B. 局域网
- C. 城域网
- D. 广域网

9. 中央处理器是由运算器和_____组成的。

- A. 存储器
- B. 输入设备
- C. 输出设备
- D. 控制器

10. _____不是系统软件。

- A. Android
- B. Windows
- C. Linux
- D. Photoshop

11. _____是衡量系统传输能力的主要指标，一般用波特率或者比特率来表示。

- A. 传输速率
- B. 带宽
- C. 波长
- D. 差错率

12. _____不属于无线网传输数据的介质。

- A. 微波线路
- B. 卫星
- C. 红外线
- D. 双绞线

13. _____不属于路由器的主要功能。

- A. 数字信号转换成模拟信号
- B. 连通不同网络
- C. 选择信息传送的线路
- D. 利用路由表选择路径

14. IP 地址 172. 10. 10. 10 属于_____类 IP 地址。

- A. A
- B. B
- C. C
- D. D

15. 使用浏览器上网浏览网站信息，使用的是互联网的_____服务。

- A. FTP
- B. WWW
- C. 电子邮件
- D. 移动引擎

16. 防火墙用于将 Internet 和内部网络隔离，_____。

- A. 是防止 Internet 火灾的硬件设施
- B. 是网络安全的软件和硬件设施
- C. 是保护线路不受破坏的软件和硬件设施
- D. 是起抗电磁干扰作用的硬件设施

17. _____伪装成正常文件，通过隐蔽的手段获取运行权，然后盗取用户的隐私信息，或者进行恶意行为。

- A. 蠕虫病毒
- B. 引导型病毒
- C. 木马病毒
- D. 宏病毒

18. Python 中，直接进入本层下一轮循环的语句是_____。

- A. next
- B. exit
- C. continue
- D. break

19. _____不是 Python 的关键字。

- A. if
- B. while
- C. for
- D. keyword

20. _____是指机器学习算法对新样本的适应能力。

- A. 模型测试
- B. 泛化能力
- C. 过拟合
- D. 模型训练

21. 2016 年 3 月，人工智能程序_____在韩国首次以 4:1 的比分战胜人类围棋冠军李世石。

- A. AlphaGo
- B. Deepblue
- C. DeepMind
- D. AlphaGo Zero

22. Python 提供了_____循环和 while 循环。

- A. for
- B. loop
- C. until
- D. do

23. _____不属于典型的人工智能研究。

- A. 人工神经网络
- B. 自动控制
- C. 自然语言处理
- D. 专家系统

24. 产生式系统的推理不包括_____。

- A. 正向推理
- B. 逆向推理
- C. 双向推理
- D. 简单推理

25. 在人工智能中，问题求解、创作、推理预测被认为是在_____的层次。

- A. 认知智能
- B. 感知智能
- C. 认感智能
- D. 行为智能

二、是非题

(本大题 5 道小题，每空 1 分，共 5 分)。以下陈述正确的选择“是”，错误的选择“否”

1. 在计算机中 MB 表示兆字节，1MB=1024B。

- 是
- 否

2. 屏幕保护程序是为了保护显示器而设计的一种专用程序。它是为了防止因无人操作而使显示器长时间显示同一个画面，导致老化而缩短显示器寿命。

- 是
- 否

3. 信号是数据在传输过程中的表现形式，分为模拟信号和电子信号。

- 是

否

4. Python 中，函数定义的代码块以 func 关键字开头，后接函数名和参数等。

是

否

5. 神经网络研究属于人工智能的连接主义学派。

是

否

三、操作题

(一) 文件管理 (共 6 分)

1. 在C:\KS文件夹下创建MYEXE子文件夹，通过搜索查找文件calc.exe，将其复制到所创建的MYEXE子文件夹中；将C:\素材\test.rar压缩包中的文件pic.jpg解压到C:\KS\MYEXE子文件夹中；设置MYEXE子文件夹属性为“只读”。

2. 在C:\KS文件夹中创建名为“画图”的快捷方式，指向Windows系统文件夹中的应用程序“mspaint.exe”，设置运行方式为“最大化”，并指定快捷键为“<Ctrl>+<Alt>+T”。

(二) 数据处理 (共 20 分)

1. 文字信息处理 (10分)

打开C:\KS\word.docx文件，请参照样张，按要求进行编辑和排版，将结果以原文件名保存在C:\KS文件夹中。

(1) 按样张创建“自动目录1”样式的二级目录。为标题添加尾注“本文摘自百度百科”，为“传播途径”添加脚注“新型冠状病毒主要传播途径是呼吸道飞沫传播和接触传播”。设置页眉为居中的“新冠病毒”，在页面下方居中插入页码，形式为“第X页，共Y页”（其中X和Y是数字）。

(2) 设置标题“2019新型冠状病毒”居中、三号、繁体、字符间距加宽为3磅，采用第3行第2列的文本效果。将正文中“三道防线”下的三段内容分成等宽三栏，并加栏间分隔线，首字下沉2行。

(3) 按样张插入C:\素材\nCoV.png图片，水平翻转并设置“冲蚀”颜色效果，衬于文字下方，并适当调整大小，放置于文档合适位置。

2. 电子表格处理 (10分)

打开C:\KS\excel.xlsx文件，请按要求进行处理，将结果以原文件名保存在C:\KS文件夹中。（计算必须用公式，否则不计分）

(1) 在Sheet1中，在表格第一行输入标题文字“上海市8月天气情况表”，在A1:G1区域跨列居中，并设置字体为华文琥珀，字体大小为16；日期数据设置为“X月Y日”的格式（其中X、Y表示数字）。在F3:F19计算平均气温（最高气温和最低气温的平均值）。在G3:G19利用IF函数判断气温情况：最高气温高于38度（包含38度）为“酷热”，35度（包含35度）到38度之间为“炎热”，低于35度为“闷热”。

(2) 在Sheet1中, 设置"最高气温"的条件格式: 将8月最热的前5天的温度数据设置为红色字体, 将8月最凉爽的4天的温度数据设置为黄色填充。复制Sheet1中A2:E19数据到Sheet2中, 在Sheet2中, 筛选出刮"西南风"和"东北风"并且除"多云"天气以外的所有数据信息。将Sheet1中A2:E8数据复制后转置粘贴到Sheet3中A1起始的位置, 使得行列互换, 并自动调整列宽。

(3) 在Sheet1中的H3:N19区域, 为"多云"天气制作最高气温数据列的"带数据标记的折线图", 套用"样式3"的图表样式, 并显示8月16日最高气温的数据标签, 如样张所示。

(三) 网络应用基础 (共 4 分)

1. 用浏览器打开C:\素材\网页E.htm, 将该网页以PDF格式保存在C:\KS文件夹中, 文件名为WYE.pdf。

2. 在C:\KS文件夹中创建IP.txt, 将当前计算机的WindowsIP配置、所有以太网适配器的信息粘贴在内, 每个信息独占一行。

(四) 人工智能编程基础 (共 5 分)

打开C:\KS文件夹下的程序文件4_1.py, 按下列要求完成程序, 并将结果以原文件名保存在C:\KS文件夹中。

程序实现以下功能:

1. 用户输入体重和身高。
2. 将体重和身高转换为浮点数。
3. 计算 BMI 指数值, $BMI = \text{体重 (千克)} / \text{身高}^2 (\text{米}^2)$ 。
4. 输出 BMI 指数值。
5. 根据 BMI 指数值输出用户的肥胖情况:
 - 如果 BMI 小于 19, 则显示偏瘦;
 - 如果大于等于 19 且小于 24, 则显示正常;
 - 如果大于等于 24, 则显示偏胖。

请按代码中的注释提示, 在横线处补全合适的程序代码, 程序运行结果如下图所示。注意: 考生只可补全代码, 不可修改或删除横线处以外任何代码。

请输入体重 (千克): 60 请输入身高 (米): 1.70 您的BMI指数是20.761, 正常	请输入体重 (千克): 80 请输入身高 (米): 1.65 您的BMI指数是29.385, 偏胖	请输入体重 (千克): 50 请输入身高 (米): 1.80 您的BMI指数是15.432, 偏瘦
---	---	---

程序代码如下:

```
#用户输入体重和身高
weight=input("请输入体重 (千克): ")      #1分 输入体重
height=input("请输入身高 (米): ")

weight=float(weight)                        #1分 体重转型为浮点数
height=float(height)

#计算 BMI 指数值
```

```

BMI=weight/(height*height) #1分 计算 BMI 指数值

print("您的 BMI 指数是%0.3f"%BMI) #1分 输出 BMI 指数值

#根据 BMI 指数值输出用户的肥胖情况
if BMI < 19: #1分 当 BMI < 19 时, 显示偏瘦
    print("偏瘦")
elif BMI < 24:
    print("正常")
else:
    print("肥胖")

```

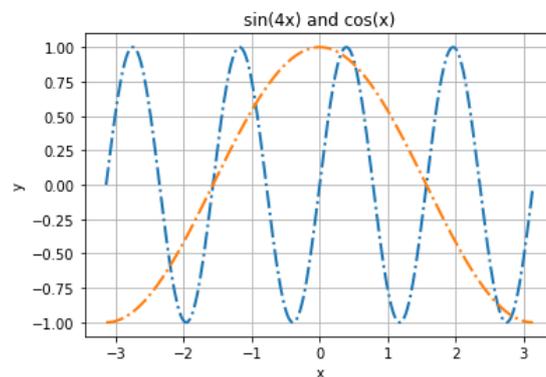
(五) 人工智能数据处理 (共 10 分)

打开C:\KS文件夹下的程序文件5_1.py, 按下列要求完成程序, 并将结果以原文件名保存在C:\KS文件夹中。

程序绘制函数 $\sin(4x)$ 和 $\cos(x)$ 在 $[-\pi, \pi]$ 上的图形, 具体要求如下:

1. 导入 matplotlib 和 numpy 库
2. 准备 $\sin(4x)$ 和 $\cos(x)$ 的绘图数据, 设置线条样式为 "-.", 宽度为 2
3. 设置标题为 "sin(4x) and cos(x)"
4. 设置坐标轴标签, 绘制 $\sin(4x)$ 和 $\cos(x)$ 曲线, 并显示网格线
5. 显示图像

请按代码中的注释提示, 在横线处补充合适的程序代码, 程序运行结果如下图所示。注意: 考生只可补全代码, 不可修改或删除横线处以外任何代码。



程序代码如下:

```

import matplotlib.pyplot as plt #2分 导入 pyplot 库并重命名
import numpy as np

x = np.arange(-np.pi,np.pi,0.01)
y1 = np.sin(4*x)
y2 = np.cos(x)
plt.rcParams['lines.linestyle']='-.'

```

<code>plt.rcParams['lines.linewidth']=2</code>	#2分 线条宽度为2
<code>plt.title('sin(4x) and cos(x)')</code>	#2分 设置标题为“sin(4x) and cos(x)”
<code>plt.xlabel('x')</code>	
<code>plt.ylabel('y')</code>	
<code>plt.plot(x,y1)</code>	
<code>plt.plot(x,y2)</code>	
<code>plt.grid()</code>	#2分 显示网格线
<code>plt.show()</code>	#2分 显示图像

(六) 机器学习 (共 15 分)

打开C:\KS文件夹下的程序文件6_1.py，按下列要求完成程序，并将结果以原文件名保存在C:\KS文件夹中。

程序实现以下功能：

1. 读取鸢尾花数据集，切分数据集，测试集所占比例为 30%。
2. 对训练集进行标准化拟合和转换，对测试集进行标准化转换。
3. 基于 KNN 算法进行分类，利用训练集的特征数据和标签数据进行模型拟合。
4. 对测试集的特征数据进行类别预测。
5. 根据测试集的标签数据，以及预测结果，计算并显示主要分类指标的文本报告。

请从以下选项中选择正确的代码填入程序中相应的横线处，补全程序。

- A、`ss.fit_transform(X_train)`
- B、`knc.predict(X_test)`
- C、`data, target, test_size=0.7`
- D、`ss.transform(X_train)`
- E、`classification_report(y_test, y_predict,`
- F、`data, target, test_size=0.3`
- G、`knc.predict(X_train)`
- H、`knc.fit(X_train, y_train)`
- I、`report(y_test, y_predict,`
- J、`knc.fit(X_test, y_test)`

程序运行结果如下图所示。注意：考生只可补全代码，不可修改或删除横线处以外任何代码。

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	15
versicolor	0.84	1.00	0.91	16
virginica	1.00	0.79	0.88	14
accuracy			0.93	45
macro avg	0.95	0.93	0.93	45
weighted avg	0.94	0.93	0.93	45

程序代码如下：

```
#导入库
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report #导入用于显示主要分类指标文本报告的函数

#利用 load_iris 读取鸢尾花数据集
iris = load_iris()
data = iris.data
target = iris.target
X_train, X_test, y_train, y_test = train_test_split(data, target, test_size=0.3, random_state=8) #3 分切分数据集，测试集占比 30%

#标准化数据
ss = StandardScaler()
X_train = ss.fit_transform(X_train) #3 分 对训练集拟合和转换
X_test = ss.transform(X_test)

#使用 K 近邻分类器对测试数据进行类别预测
knc = KNeighborsClassifier()
knc.fit(X_train, y_train) #3 分 模型拟合
y_predict = knc.predict(X_test) #3 分 模型预测
#显示主要分类指标的文本报告
report = classification_report(y_test, y_predict, target_names=iris.target_names) #3 分
print(report)
```

（七）深度学习（共 10 分）

打开C:\KS文件夹下的程序文件7_1.py，阅读和分析程序，按下列要求完成题目，并将结果以原文件名保存在C:\KS文件夹中。

程序通过神经网络对 MNIST 数据集进行分类训练和模型评估。请针对程序中 5 处【】所在的代码行，从以下选项中选择对该行恰当的代码解释，并将编号填入【】内，如【A】，编号不区分大小写。

- A、打印数据集的概况。
- B、创建 Tensorflow 网络模型。
- C、载入 MNIST 数据集，并将训练集的特征数据和标签数据保存在 X_train 和 y_train，将测试集的特征数据和标签数据保存在 X_test 和 y_test。
- D、打印模型的概况。
- E、模型训练，代数达到 10 时结束训练。
- F、添加隐藏层，该层包含 100 个神经元，输入维度为 28*28，激活函数为 relu。
- G、添加隐藏层，该层输入维度为 100，包含 28*28 个神经元，激活函数为 relu。
- H、载入 MNIST 数据集，并将测试集的特征数据和标签数据保存在 X_train 和 y_train，将训练集的特征数据和标签数据保存在 X_test 和 y_test。
- I、创建神经网络模型。
- J、模型训练，误差达到 10 时结束训练。

程序运行结果如下图所示。注意：此题无需运行和调试程序，仅做阅读和分析。

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 100)	78500
Output (Dense)	(None, 10)	1010

Total params: 79,510
Trainable params: 79,510
Non-trainable params: 0

None
Train on 60000 samples
Epoch 1/10
60000/60000 [=====] - 6s 104us/sample - loss: 0.2695 - accuracy: 0.9232
Epoch 2/10
60000/60000 [=====] - 6s 100us/sample - loss: 0.1226 - accuracy: 0.9643
Epoch 3/10
60000/60000 [=====] - 6s 94us/sample - loss: 0.0864 - accuracy: 0.9741
Epoch 4/10
60000/60000 [=====] - 5s 86us/sample - loss: 0.0656 - accuracy: 0.9812
Epoch 5/10
60000/60000 [=====] - 5s 86us/sample - loss: 0.0509 - accuracy: 0.9848
Epoch 6/10
60000/60000 [=====] - 5s 91us/sample - loss: 0.0416 - accuracy: 0.9869
Epoch 7/10
60000/60000 [=====] - 5s 88us/sample - loss: 0.0335 - accuracy: 0.9897
Epoch 8/10
60000/60000 [=====] - 5s 87us/sample - loss: 0.0287 - accuracy: 0.9913
Epoch 9/10
60000/60000 [=====] - 5s 86us/sample - loss: 0.0233 - accuracy: 0.9926
Epoch 10/10
60000/60000 [=====] - 5s 86us/sample - loss: 0.0194 - accuracy: 0.9941
10000/10000 [=====] - 1s 63us/sample - loss: 0.0928 - accuracy: 0.9732

程序代码如下：

```
#导入 TensorFlow
import tensorflow as tf

mnist = tf.keras.datasets.mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

#2分 [C]

```
#利用 reshape 函数转换数字图像
```

```
X_train_reshape = X_train.reshape(X_train.shape[0], 28*28)
```

```
X_test_reshape = X_test.reshape(X_test.shape[0], 28*28)
```

```
#归一化数字图像
```

```
X_train_norm, X_test_norm = X_train_reshape / 255.0, X_test_reshape / 255.0
```

```
model = tf.keras.models.Sequential()
```

#2分 **[I]**

```
model.add(tf.keras.layers.Dense(100,input_dim=28*28,activation='relu'))
```

#2分 **[F]**

```
model.add(tf.keras.layers.Dense(10,activation='softmax',name='Output'))
```

```
print(model.summary())
```

#2分 **[D]**

```
model.compile(optimizer='adam',
```

```
              loss='sparse_categorical_crossentropy',
```

```
              metrics=['accuracy'])
```

```
model.fit(X_train_norm, y_train, epochs=10)
```

#2分 **[E]**

```
model.evaluate(X_test_norm, y_test, verbose=1)
```