1. 有Python程序段如下所示：

import jieba

str="杰尼龟在学信息"

ls = list(jieba.cut(str)) #分词结果为[‘杰尼龟’,’在’,’学’，’信息’]

for i in ls[::-1]:

print(i,end=' ')

该程序段输出结果为 （ ）

A．杰尼龟 在 学 信息

B.杰尼龟在学信息

C.信息 学 在 杰尼龟

D.信息学在杰尼龟

1. 有Python程序段如下所示，请补充划线处语句：

import collections

import matplotlib

import jieba

import wordcloud as wc

import numpy as np

from PIL import Image

wcg = wc.WordCloud(background\_color = "white",font\_path='assets/deng.ttf')

text = open('data/绿.txt',encoding='utf-8').read()

seg\_list=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

f=collections.Counter(seg\_list)

wcg.fit\_words(f)

wcg.to\_file('b.png')

1.jieba用来切词，方法是查内置词典

2.matplotlib绘图，功能强大

3.wordcloud做词云

4.figure( ),plot( ),bar( ),barh( ),scatter( ),title( ),x/ylim( ),x/ylabel( ),legend( ),show( )

当Python碰上图——Matplotlib，jieba

3.下列代码中能绘制散点图的是 （ ）

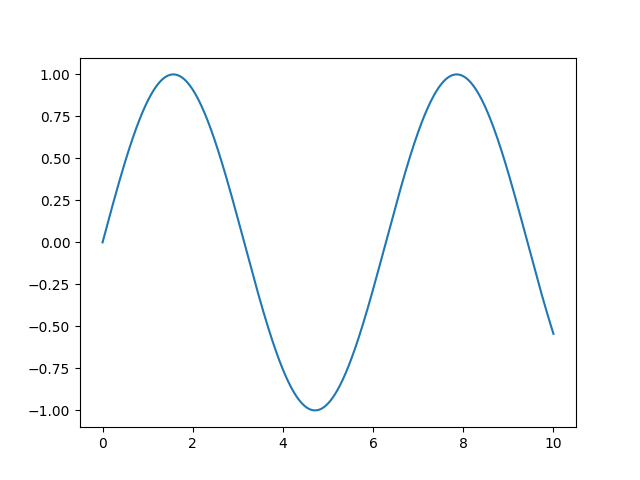
A. plt.scatter( x , y )

B. plt.plot( x , y )

C. plt.legend( )

D. plt.xlabel( ‘散点图’ )

4.杰尼龟在学数学，想利用Python绘制y=sin(x)的图象，如图所示



import matplotlib.pyplot as plt

import numpy as np

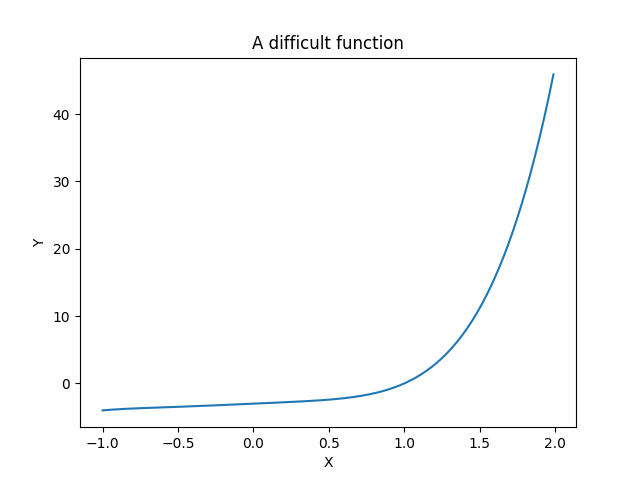
x = np.linspace( 0 , 10 ,1000 )

y = np.sin( x )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

———————————

5.杰尼龟大吃一惊，原来Python这么好用，于是它又打算让你帮忙搞一个复杂点的函数。请帮它填空。



import matplotlib.pyplot as plt

import numpy as np

x = np.arange( -1 , 2 , 0.01 )

y=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

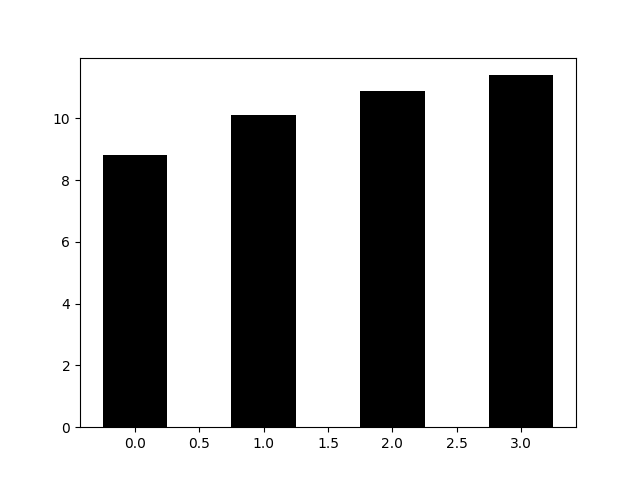
plt.plot( x , y )

plt.title('A difficult function')

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

plt.ylabel( 'Y' )

plt.show( )

6.杰尼龟最近在种盆栽，它对盆栽中的某一枝条做好标记，记录28天内该枝条的生长情况，每周记录一次，四次记录结果分别为8.8cm , 10.1cm , 10.9cm , 11.4 cm。它用Python制作了柱形图

import matplotlib.pyplot as plt

import numpy as np

y = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

index = np.arange(4)

plt.\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_,height=y,color="black",width=0.5)

plt.show()

7.图像的存储和处理在计算机内部的是采用二进制的形式，一幅图像常常由分辨率、颜色深度的方式来存储，Python的PIL模块能够直接读取图像中的像素点的颜色信息，一般采用RGB（红：0~255，绿：0~255，蓝：0~255）的方式表示颜色。现有一副红字白底的图像1.bmp，基本信息如下：



程序段1

import PIL import Image

import matplotlib.pyplot as plt

import numpy as np

img=np.array(Image.open('1.bmp').convert("L"))

rows,cols=img.shape #获取图像的行数和列数

for i in range(rows):

for j in range(cols):

if img[i,j]>128:

img[i,j]=0

else:

img[i,j]=1

plt.figure("1.bmp")

plt.imshow(img.camp="gray")

plt.axis("off")

plt.show( )

（1）程序段1运行过程中，变量j的取值范围是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(2)程序段2调试时出现“IndexError : index 128 is out of bounds for axis 1 with size 128”的提示，则上述程序出现这个错误的语句是（ ）

A. rows,cols=img.shape B. for i in range( rows )

C. img[i,j],img[i,cols-j]=img[i,cols-j],img[i,j] D. plt.imshow(img,cmap=”gray”)

应该修改为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

程序段2

import PIL import image

import matplotlib.pyplot as plt

import numpy as np

img = np.array(Image.open('1.bmp').convert("L"))

rows,cols=img.shape #获取图像的行数和列数

for i in range(rows):

for j in range(cols//2):

img[i,j],img[i,cols-j]=img[i,cols-j],img[i,j]

plt.figure("1.bmp")

plt.imshow(img.camp="gray")

plt.axis("off")

plt.show( ) 程序段的效果是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

小题特训：

import random

a=[0,0,0,0,0]

flag=1

i=0

while i<5:

b=random.randint(1,100)

if flag>0 or b>50:

a[i]=flag\*b

i+=1

flag=-flag

print(a)

执行程序后，输出的列表a可能是 （ ）

A.[5,-53,23,-65,38] B.[-67,64,-91,78,-31]

C.[4,-69,0,-84,34] D.[60,-30,71,-29,88]