

第七次作业内容

0 / 2

编程题

1. # 请将"_"处替换成能实现注释中功能的代码。

示例:

#定义列表lst, 值为 1, 3, 4

lst = _

根据注释中说明可知, 此处"_"应替换成[1, 3,4],即有

lst = [1, 3,4]

import numpy as np

def fact():

定义answer, 用于检查结果是否正确,用于批改作业

answer=[]

创建一维的ndarray对象arr1, 内有元素1, 2, 3, 4, 5, 6, 7, 8, 9要求使用arange()函数

arr1 = _

将结果添加到answer, 用于检查

answer.append(arr1.copy())

将arr1转换成3*3的矩阵arr2

arr2 = _

将结果添加到answer, 用于检查

answer.append(arr2.copy())

使用linspace()函数, 生成首位是0, 末位是10, 含5个数的等差数列arr3,元素类型为float

arr3 = _

将结果添加到answer, 用于检查

answer.append(arr3.copy())

创建3*4的全1矩阵arrOnes, 元素类型为int

arrOnes = _

将结果添加到answer, 用于检查

answer.append(arrOnes.copy())

创建3*4的全0矩阵arrZeros, 元素类型为int

arrZeros = _

```
# 将结果添加到answer, 用于检查  
answer.append(arrZeros.copy())
```

```
# 创建3阶单位矩阵arrUnit, 元素类型为int  
arrUnit = _  
# 将结果添加到answer, 用于检查  
answer.append(arrUnit.copy())
```

```
# 创建一个3*3的矩阵matrix1, 内有元素[[1,3,3],[6,5,6],[9,9,9]],元素类型为int  
matrix1 = _  
# 将结果添加到answer, 用于检查  
answer.append(matrix1.copy())
```

```
# 获取矩阵matrix1的逆为matrix2  
matrix2 = _  
# 将结果添加到answer, 用于检查  
answer.append(matrix2.copy())
```

```
# 打印矩阵matrix1中的最大值  
maxOfMatrix1 = _
```

```
# 打印矩阵matrix1每一列的最大值  
ColumnMax = _  
# 将结果添加到answer, 用于检查  
answer.append(ColumnMax.copy())
```

```
# 打印矩阵matrix1每一行的平均值  
LineMean = _  
# 将结果添加到answer, 用于检查  
answer.append(LineMean.copy())
```

```
# 打印矩阵matrix1每一列的方差  
variance = _  
# 将结果添加到answer, 用于检查  
answer.append(variance.copy())
```

```
# 截取矩阵matrix1的第1, 2行, 存到matrix3  
matrix3 = _  
# 将结果添加到answer, 用于检查  
answer.append(matrix3.copy())
```

```
# 截取矩阵matrix1的第1, 2行, 第2, 3列,存到matrix4  
matrix4 = _  
# 将结果添加到answer, 用于检查  
answer.append(matrix4.copy())
```

```
# 截取矩阵matrix1中大于3的元素  
maxList = _  
# 将结果添加到answer, 用于检查
```

```
answer.append(maxList.copy())
```

```
return answer,maxOfMatrix1
```

你的答案：

```
import numpy as np

def fact():
    # 定义answer，用于检查结果是否正确,用于批改作业
    answer=[]

    # 创建一维的narray对象arr1，内有元素1，2，3，4，5，6，7，8，9要求使用arange()
    函数
    arr1 = np.arange(1, 10)
    # 将结果添加到answer，用于检查
    answer.append(arr1.copy())

    # 将arr1转换成3*3的矩阵arr2
    arr2 = arr1.reshape(3, 3)
```

2. # 请将"_"处替换成能实现注释中功能的代码。

示例：

#定义列表lst，值为 1，3，4

lst = _

根据注释中说明可知，此处"_"应替换成[1, 3,4],即有

lst = [1, 3,4]

```
import numpy as np
```

```
def fact():
```

定义answer，用于检查结果是否正确,用于批改作业

```
answer=[]
```

创建一维的narray对象arr1，内有元素1，2，3，4，5，6，7，8，9要求使用arange()函数

```
arr1 = _
```

将结果添加到answer，用于检查

```
answer.append(arr1.copy())
```

将arr1转换成3*3的矩阵arr2

```
arr2 = _
```

将结果添加到answer，用于检查

```
answer.append(arr2.copy())
```

使用linspace()函数，生成首位是0，末位是10，含5个数的等差数列arr3,元素类型为float

```
arr3 = _
```

将结果添加到answer，用于检查

```
answer.append(arr3.copy())
```

创建3*4的全1矩阵arrOnes，元素类型为int

```
arrOnes = _
```

```
# 将结果添加到answer, 用于检查
answer.append(arrOnes.copy())

# 创建3*4的全0矩阵arrZeros, 元素类型为int
arrZeros = _
# 将结果添加到answer, 用于检查
answer.append(arrZeros.copy())

# 创建3阶单位矩阵arrUnit, 元素类型为int
arrUnit = _
# 将结果添加到answer, 用于检查
answer.append(arrUnit.copy())

# 创建一个3*3的矩阵matrix1, 内有元素[[1,3,3],[6,5,6],[9,9,9]],元素类型为int
matrix1 = _
# 将结果添加到answer, 用于检查
answer.append(matrix1.copy())

# 获取矩阵matrix1的逆为matrix2
matrix2 = _
# 将结果添加到answer, 用于检查
answer.append(matrix2.copy())

# 打印矩阵matrix1中的最大值
maxOfMatrix1 = _

# 打印矩阵matrix1每一列的最大值
ColumnMax = _
# 将结果添加到answer, 用于检查
answer.append(ColumnMax.copy())

# 打印矩阵matrix1每一行的平均值
LineMean = _
# 将结果添加到answer, 用于检查
answer.append(LineMean.copy())

# 打印矩阵matrix1每一列的方差
variance = _
# 将结果添加到answer, 用于检查
answer.append(variance.copy())

# 截取矩阵matrix1的第1, 2行, 存到matrix3
matrix3 = _
# 将结果添加到answer, 用于检查
answer.append(matrix3.copy())

# 截取矩阵matrix1的第1, 2行, 第2, 3列,存到matrix4
matrix4 = _
# 将结果添加到answer, 用于检查
```

```
answer.append(matrix4.copy())

# 截取矩阵matrix1中大于3的元素
maxList = _
# 将结果添加到answer, 用于检查
answer.append(maxList.copy())

return answer,maxOfMatrix1
```

你的答案:

```
import numpy as np

def fact():
    # 定义answer, 用于检查结果是否正确, 用于批改作业
    answer=[]

    # 创建一维的narray对象arr1, 内有元素1, 2, 3, 4, 5, 6, 7, 8, 9要求使用arange()
    函数
    arr1 = np.arange(1, 10)
    # 将结果添加到answer, 用于检查
    answer.append(arr1.copy())

    # 将arr1转换成3*3的矩阵arr2
    arr2 = arr1.reshape(3, 3)
```

返回