#!/usr/bin/env python

# coding: utf-8

# In[ ]:

# numpy assignment 23/01/2023

# due on 30/01/2023 EOD

# formlink to upload the doc: https://forms.office.com/r/kY6avu9rJN

# In[ ]:

# Q0. How to create an empty and a full NumPy array(filled with 5) of size (5,4)?

# In[115]:

import numpy as np

print(np.ones(20).reshape(5,4)\*5)

# In[ ]:

# Q1. Find the most frequent value in a NumPy array [1,2,3,4,5,1,2,1,1,1]

# In[38]:

x=np.array([1,2,3,4,5,1,2,1,1,1])

print(np.bincount(x).argmax())

# In[ ]:

# Q2: Construct a border of 0s around 2D matrix

# input\_arr = [[1,2],

# [3,4]]

# res\_arr = [

# [0,0,0,0],

# [0,1,2,0],

# [0,3,4,0],

# [0,0,0,0],

# ]

# In[4]:

input\_arr=np.array([[1,2],[3,4]])

res\_arr=np.zeros(16).reshape(4,4)

res\_arr[1:3,1:3]=input\_arr

print(res\_arr)

# In[ ]:

# Q3. Combine these two arrays(arr1 and arr2) into this resultant array(res\_arr):

# arr1 = [[1, 2],

# [3, 4.]]

# arr2 = [[5, 6, 7],

# [8, 9, 10]]

# res\_arr = [[ 1., 2., 5., 6., 7.],

# [ 3., 4., 8., 9., 10.]

# In[16]:

arr1=np.array([[1,2],[3,4]])

arr2=np.array([[5,6,7],[8,9,10]])

res\_arr=np.concatenate((arr1,arr2),axis=1)

print(res\_arr)

# In[ ]:

# Q4. Find the number of occurrences of a sequence in a NumPy array?

# arr = numpy.array([[9, 4, 5, 9, 4],

# [4, 7, 8, 4, 7],

# [2, 9, 9, 4, 7]])

# Counting sequence 4, 7

# In[28]:

arr=np.array([[9, 4, 5, 9, 4],[4, 7, 8, 4, 7],[2, 9, 9, 4, 7]])

count=0

for row in arr:

for i in range(len(row)-1):

if row[i:i+2].tolist()==[4, 7]:

count=count+1

print(count)

# In[ ]:

# Q5. Find the most frequent value in a NumPy array? [[3,2,1],[5,4,3]]

# In[39]:

arr=np.array([[3,2,1],[5,4,3]])

flat\_arr=arr.flatten()

print(np.bincount(flat\_arr).argmax())

# In[ ]:

# Q6. Find the 3rd largest value and it's index in numpy array?

# arr: [2 0 1 5 4 1 9,4]

# res: 3rd largest value is 4 and its index are [4,7]

# In[54]:

arr=np.array([2,0,1,5,4,1,9,4])

uniq\_arr=np.unique(arr)

sort\_arr=np.sort(uniq\_arr)

third\_large=sort\_arr[-3]

indices=np.where(third\_large==arr)

print("3rd largest value is {0} and it's index are {1}".format(third\_large,indices[0]))

# In[ ]:

# Q7. Find euclidean distance between two numpy arrays?

# point1 = np.array((1, 5, 6, 2, 3))

# point2 = np.array((3, 7, 11, 4, 1))

# In[55]:

point1 = np.array((1, 5, 6, 2, 3))

point2 = np.array((3, 7, 11, 4, 1))

distance = np.linalg.norm(point1 - point2)

print(distance)

# In[ ]:

# Q8. Trim the leading and/or trailing zeros from a 1-D array or sequence

# and ensure only one 0 is present before and after the values?

# input\_arr = [0,0,0,1,3,4,5,0,6,0,0,0,0,0,0,0]

# output\_arr = [0,1,3,4,5,0,6,0]

# In[97]:

arr=np.array([0,0,0,1,3,4,5,0,6,0,0,0,0,0,0,0])

i=0

while arr[i]==0 and arr[i+1]==0:

arr=np.delete(arr,i)

i=len(arr)-1

while arr[i]==0 and arr[i-1]==0:

arr=np.delete(arr,i)

i=i-1

print(arr)

# In[ ]:

# Q9. Sorted a given complex array using the real part first, then the imaginary part.

# Original array:

# [(1+2j), (3-1j), (3-2j), (4-3j), (3+5j)]

# res:

# [1.+2.j 3.-2.j 3.-1.j 3.+5.j 4.-3.j]

# In[98]:

arr=np.array([1+2j,3-1j,3-2j,4-3j,3+5j])

print(np.sort\_complex(arr))

# In[ ]:

# Q10. Sort the following students array with class first and then height in each class?

# import numpy as np

# data\_type = [('name', 'S15'), ('class', int), ('height', float)]

# students\_details = [('James', 5, 48.5), ('Nail', 6, 52.5),('Paul', 5, 42.10), ('Pit', 6, 40.11)]

# # create a structured array

# students = np.array(students\_details, dtype=data\_type)

# In[105]:

data\_type = [('name', 'S15'), ('class', int), ('height', float)]

students\_details = [('James', 5, 48.5), ('Nail', 6, 52.5),('Paul', 5, 42.10), ('Pit', 6, 40.11)]

#create a structured array

students = np.array(students\_details, dtype=data\_type)

students = np.sort(students, order=['class', 'height'])

print(students)

# In[ ]:

# Q11. How to build an array of all combinations of two NumPy arrays?

# creating 2 numpy arrays

# array\_1 = np.array([1, 2])

# array\_2 = np.array([4, 6,4])

# # RESULT:

# [[1 4]

# [1 6]

# [1 4]

# [2 4]

# [2 6]

# [2 4]]

# In[117]:

array\_1=np.array([1,2])

array\_2=np.array([4,6,4])

result = np.array(np.meshgrid(array\_1, array\_2)).T.reshape(-1,2)

print(result)

# In[111]:

# Q13. Reverse this array?

# input\_arr: [[1,2,3,4,5],[6,7,8,9,10]]

# output\_arr: [[5,4,3,2,1],[10,9,8,7,6]]

# In[123]:

input\_arr = np.array([[1,2,3,4,5],[6,7,8,9,10]])

output\_arr = np.fliplr(input\_arr)

print(output\_arr)

# In[124]:

# Q14. write a function that generates first n numbers in hemachandra series using Binet's formula?

# In[127]:

import numpy as np

def hemachandra\_series(n):

t = (1 + np.sqrt(2)) / 2

s = (1 - np.sqrt(2)) / 2

fib = (t \*\* np.arange(1, n+1) - s \*\* np.arange(1, n+1)) / np.sqrt(2)

return np.around(fib).astype(int).tolist()

print(hemachandra\_series(10))

# In[128]:

# Q15. write a function that creates an array of lengths for each element in a string array?

# input\_arr= np.array(['hi','I','am','india'],dtype=np.str)

# output\_arr = np.array([2,1,2,5])

# In[134]:

def lengths\_of\_strings(input\_arr):

return np.array([len(s) for s in input\_arr], dtype=int)

input\_arr=np.array(['hi','I','am','india'],dtype=str)

print(lengths\_of\_strings(input\_arr))

# In[135]:

# Q16. Check whether a Numpy array contains a specified row

# arr:np.array([[1,2],[3,4],[4,5]])

# row:[3,4]

# res: True

# In[145]:

arr=np.array([[1,2],[3,4],[4,5]])

row=np.array([3,4])

res=np.all(np.isin(row,arr))

print(res)

# In[146]:

# Q17. Remove rows in Numpy array that contains non-numeric values?

# input arr: np.array([[1,2],[3,4],[4,np.nan]])

# res: np.array([[1,2][3,4]])

# In[148]:

arr = np.array([[1, 2], [3, 4], [4, np.nan]])

arr = arr[~np.isnan(arr).any(axis=1)]

print(arr)

# In[151]:

def write\_to\_file(arr, file\_name):

np.savetxt(file\_name, arr, delimiter=',', fmt='%.2f')

arr = np.array([[1, 2], [3, 4], [4, 5]])

write\_to\_file(arr, 'array.txt')

# In[152]:

# Q19. Write a function that converts a image file to numpy array

# and check how many pixels it has and how many unique pixels it has?

# In[157]:

import numpy as np

from PIL import Image

def analyze\_image(file\_path):

img = Image.open(file\_path)

arr = np.array(img)

num\_pixels = arr.shape[0] \* arr.shape[1]

unique\_pixels, counts = np.unique(arr, return\_counts=True)

num\_unique\_pixels = len(unique\_pixels)

return num\_pixels, num\_unique\_pixels

num\_pixels, num\_unique\_pixels = analyze\_image('img.gif')

print('Number of pixels:', num\_pixels)

print('Number of unique pixels:', num\_unique\_pixels)

# In[154]:

# Q20. convert the following dictionary to numpy array?

# input\_dict = {1:2, 2:3, 3:4}

# output\_arr = np.array([[1,2],[2,3],[3,4]])

# In[156]:

import numpy as np

input\_dict = {1:2, 2:3, 3:4}

output\_arr = np.array([[k, v] for k, v in input\_dict.items()])

print(output\_arr)