Kevin Caton-Largent

kcaton-largent

100067438

Lab 5 section 03L Mondays

**Exercise 1**

|  |
| --- |
| #a  s = ' remote sensing '  s1 = s.\_\_len\_\_()  print s1  results:  18 |

|  |
| --- |
| #b  s = ' remote sensing '  s2 = s.strip(' ')  print s2  results:  remote sensing |

|  |
| --- |
| #c  s = ' remote sensing '  s2 = s.strip(' ')  s3 = s2.split(' ', 2)  print s3  results:  ['remote', 'sensing'] |

|  |
| --- |
| #d  s = ' remote sensing '  s4 = s.swapcase()  print s4  results:  REMOTE SENSING |

**Exercise 2**

|  |
| --- |
| results = [] #initialize an empty list  Transect = [] # another empty list  rf = open('Transect.csv','r')  header = rf.readline().split('\n',3) #reads the first line of the file as a header  # part A  sum\_density = 0  for i in range(0,128):  line = rf.readline() #read one line in file  line = line.rsplit (',', 2)  results.append(line[1])  for i in range(0,128):  sum\_density += int( results[i])    avg\_density = sum\_density/128 |

|  |
| --- |
| Results:  2 |

|  |
| --- |
| # part B writing to Patch\_Gap file  # setup for part B  wf = open('Patch\_Gap.csv','w')  wf.write (header[0] + ',Patch\_Gap\n')  position = rf.seek(0,0)  header = rf.readline().split('\n',3)  for i in range(0,128):  line = rf.readline().split('\n',3)  if int(results[i]) >= avg\_density:  wf.write(line[0] + ',1\n')  else:  wf.write(line[0] + ',0\n')  rf.close()  wf.close() |

**Exercise 3**

|  |
| --- |
| results = [] #initialize an empty list  Transect = [] # another empty list  rf = open('Transect.csv','r')  header = rf.readline().split(',') #reads the first line of the file as a header  # part A  sum\_density = 0  for line in rf:  line = line.rsplit (',', 2)  results.append(line[1])  for i in range(0,128):  sum\_density += int( results[i])    rf.close()  avg\_density = sum\_density/128  # part B writing to Patch\_Gap file  # setup for part B  rf = open('Transect.csv','r')  header = rf.readline().split('\n',3)  wf = open('Patch\_Gap2.csv','w')  wf.write (header[0] + ',Patch\_Gap\n')  rf.close()  # part B  rf = open('Transect.csv','r')  header = rf.readline().split('\n',3)  i = 0  for line in rf:  line = line.split('\n',3)  if int(results[i]) >= avg\_density:  wf.write(line[0] + ',1\n')  else:  wf.write(line[0] + ',0\n')  i += 1  rf.close()  wf.close() |

**Exercise 4**

|  |
| --- |
| import numpy as np  import re  rf = open ('Image.asc', 'r')  size = rf.readline()  size = re.split('[^0-9]\*', size)  size.pop(-1) # the split splits it into 4 elements 2 are empty space  size.pop(0) # these pops are to remove these extra elements so only the dimensions are left  x = int(size[0])  y = int(size[1])  myArr = np.empty((y,x),int) #create an empty 2-dimensional int array of length 50  numList = []  for num in rf:  num = num.split()  numList.append(num)  rf.close()  m = 0  n = 0  for i in range(0,y):  for j in range (0,x):  myArr[i][j] = int(numList[i][j])  print myArr  print np.max(myArr)  print np.min(myArr) |

|  |
| --- |
| Results:  114  44 |

**Exercise 5**

|  |
| --- |
| import numpy as np  # part B  def get\_SI(arr):  N = 0.0  H = 0.0  for i in range (0,len(arr)):  N = N + float(arr[i])  for i in range(0,len(arr)):  H = H + ((arr[i]/N) \* np.log(arr[i]/N))  return (-H)    # part A  rf = open ('Species.csv')  header = rf.readline().split(',')  inNum = np.empty((6),int)  numList = []  for num in rf:  num = num.split(',')  numList.append(int(num[1]))  rf.close()  m = 0  n = 0  for i in range(0,6):  inNum[i] = int(numList[i])  arr = np.array([10,20,40])  print get\_SI(arr) |

|  |
| --- |
| Results:  0.955699891113 |

**Exercise 6**

|  |
| --- |
| import math  import itertools  def distance (p0,p1):  return math.sqrt((p0[0] - p1[0])\*\*2 + (p0[1] - p1[1])\*\*2)  rf = open ('Points.csv', 'r')  header = rf.readline().split(',')  dlist = []  for line in rf:  line = line.split(',',3)  x,y = int(line[1]),int(line[2])  dlist += [(x,y)]  distances = []  for p0, p1 in itertools.combinations(dlist,2):  distances.append(distance(p0,p1))  print distances  rf.close() |