```python
'''
*************************************************************************************
Assignment: U9A4 NumPy 2D Array - Lookup Table

Description: A program that determines a 2-dimensional array of the volume correction
factor (VCF) data with specific rows and columns according to the given text file using a
function called readData(filename), as well as a float value for the volume fixed to 15
degrees Celsius using a function called convertVolume(vol, temp, lookup).

Author: Anantpal Singh Matharoo
Student Number: 623012

Date Start: November 01, 2020
Date Completed: November 03, 2020
*************************************************************************************
'''

import numpy as np

# readData(filename: str):ndarray

def readData(filename):
  """readData function that returns the VCF data from the text file in a 2-D array"""

  # opens the specified fuelLookup.txt file for reading input and converts to 1 list
  inputData = open(filename)
  lst = []
  inputData.readline()
  for i in inputData:
    stringLst = i.split()
    for j in stringLst:
      lst.append(float(j))
  inputData.close()

  # compensates for empty spaces in first and last row of data to allow equal reshaping
  for i in range(2, 11):
    lst.insert(i, None)
  for i in range(9):
    lst.append(None)

  # converts list into numpy array of 2 dimension with specific # of rows and columns
  arr = np.array(lst)
  arr1 = arr.reshape(82, 11)

  return arr1


# convertVolume(vol: float, temp: float, lookup: ndarray): float

def convertVolume(vol, temp, lookup):
  """convertVolume function that returns the volume corrected to 15 degrees Celsius"""

  # determines which correct row to look at for the VCF according to the temp
  if -0 < temp < 0:
    row = 40
  elif 0 < temp < 1:
    row = 41
  else:
    tempColumn = int(temp)
    x = np.where(lookup == tempColumn)
    y = x[0]
    row = y[0]

  # determines which correct column to look at for the VCF according to the temp
  column = round(((temp - int(temp)) * 10) + 1)
  if column <= 0:
    column = abs(column) + 2

  VCF = lookup[row, column]

  return round(vol * VCF, 6)


# Code to test the readData() function and convertVolume() function

def testCode():
  # independent test for readData() function
  print("2-D Array of volume correction factor data from content of fuelLookup.txt:")
  print(readData("fuelLookup.txt"), "\n")

  # unison tests for readData() function and convertVolume() function
  print("Various volumes of fuel corrected to 15 degrees Celsius according to the average
measured temperatures of the liquid during the deliveries:")
  print(convertVolume(10000, -40, readData("fuelLookup.txt")))
  print(convertVolume(10000, -35.8, readData("fuelLookup.txt")))
  print(convertVolume(10000, -17.1, readData("fuelLookup.txt")))
  print(convertVolume(10000, -0.6, readData("fuelLookup.txt")))
  print(convertVolume(10000, 0, readData("fuelLookup.txt")))
  print(convertVolume(10000, 0.6, readData("fuelLookup.txt")))
  print(convertVolume(10000, 21.3, readData("fuelLookup.txt")))
  print(convertVolume(10000, 30.5, readData("fuelLookup.txt")))
  print(convertVolume(10000, 40, readData("fuelLookup.txt")))
  print(convertVolume(875153.67, 0.4, readData("fuelLookup.txt")))
  print(convertVolume(7688153.67, -0.4, readData("fuelLookup.txt")))
  print(convertVolume(153.67, 1.4, readData("fuelLookup.txt")))
  print(convertVolume(4, 28.4, readData("fuelLookup.txt")))


testCode()

'''
*****************************************************************************************
OUTPUT FOR THE PROGRAM:

2-D Array of volume correction factor data from content of fuelLookup.txt:
[[-40.0 1.0502 None None None None None None None None None]
 [-39.0 1.0493 1.0494 1.0495 1.0496 1.0497 1.0498 1.0499 1.05 1.05 1.0501]
 [-38.0 1.0484 1.0485 1.0486 1.0487 1.0488 1.0489 1.049 1.0491 1.0492
  1.0492]
 [-37.0 1.0475 1.0476 1.0477 1.0478 1.0479 1.048 1.0481 1.0482 1.0483
  1.0483]
 [-36.0 1.0466 1.0467 1.0468 1.0469 1.047 1.0471 1.0472 1.0473 1.0474
  1.0474]
 [-35.0 1.0457 1.0458 1.0459 1.046 1.0461 1.0462 1.0463 1.0464 1.0465
  1.0465]
 [-34.0 1.0448 1.0449 1.045 1.0451 1.0452 1.0453 1.0454 1.0455 1.0456
  1.0456]
 [-33.0 1.0439 1.044 1.0441 1.0442 1.0443 1.0444 1.0445 1.0446 1.0447
  1.0447]
 [-32.0 1.043 1.0431 1.0432 1.0433 1.0434 1.0435 1.0436 1.0437 1.0438
  1.0438]
 [-31.0 1.0421 1.0422 1.0423 1.0424 1.0425 1.0426 1.0427 1.0428 1.0429
  1.0429]
 [-30.0 1.0412 1.0413 1.0414 1.0415 1.0416 1.0417 1.0418 1.0419 1.042 1.042]
 [-29.0 1.0403 1.0404 1.0405 1.0406 1.0407 1.0408 1.0409 1.041 1.0411
  1.0411]
 [-28.0 1.0394 1.0395 1.0396 1.0397 1.0398 1.0399 1.04 1.0401 1.0402 1.0402]
 [-27.0 1.0385 1.0386 1.0387 1.0388 1.0389 1.039 1.0391 1.0392 1.0392
  1.0393]
 [-26.0 1.0376 1.0377 1.0378 1.0379 1.038 1.0381 1.0382 1.0383 1.0383
  1.0384]
 [-25.0 1.0367 1.0368 1.0369 1.037 1.0371 1.0372 1.0373 1.0373 1.0374
  1.0375]
 [-24.0 1.0358 1.0359 1.036 1.0361 1.0362 1.0363 1.0364 1.0364 1.0365
  1.0366]
 [-23.0 1.0349 1.035 1.0351 1.0352 1.0353 1.0354 1.0354 1.0355 1.0356
  1.0357]
 [-22.0 1.034 1.0341 1.0342 1.0343 1.0344 1.0344 1.0345 1.0346 1.0347
  1.0348]
 [-21.0 1.0331 1.0332 1.0333 1.0334 1.0334 1.0335 1.0336 1.0337 1.0338
  1.0339]
 [-20.0 1.0322 1.0323 1.0324 1.0324 1.0325 1.0326 1.0327 1.0328 1.0329
  1.033]
 [-19.0 1.0313 1.0314 1.0314 1.0315 1.0316 1.0317 1.0318 1.0319 1.032
  1.0321]
 [-18.0 1.0304 1.0304 1.0305 1.0306 1.0307 1.0308 1.0309 1.031 1.0311
  1.0312]
 [-17.0 1.0294 1.0295 1.0296 1.0297 1.0298 1.0299 1.03 1.0301 1.0302 1.0303]
 [-16.0 1.0285 1.0286 1.0287 1.0288 1.0289 1.029 1.0291 1.0292 1.0293
  1.0294]
 [-15.0 1.0276 1.0277 1.0278 1.0279 1.028 1.0281 1.0282 1.0283 1.0284
  1.0284]
 [-14.0 1.0267 1.0268 1.0269 1.027 1.0271 1.0272 1.0273 1.0273 1.0274
  1.0275]
 [-13.0 1.0258 1.0259 1.026 1.0261 1.0262 1.0263 1.0263 1.0264 1.0265
  1.0266]
 [-12.0 1.0249 1.025 1.0251 1.0252 1.0252 1.0253 1.0254 1.0255 1.0256
  1.0257]
 [-11.0 1.024 1.0241 1.0242 1.0242 1.0243 1.0244 1.0245 1.0246 1.0247
  1.0248]
 [-10.0 1.0231 1.0231 1.0232 1.0233 1.0234 1.0235 1.0236 1.0237 1.0238
  1.0239]
 [-9.0 1.0221 1.0222 1.0223 1.0224 1.0225 1.0226 1.0227 1.0228 1.0229 1.023]
 [-8.0 1.0212 1.0213 1.0214 1.0215 1.0216 1.0217 1.0218 1.0219 1.022 1.022]
 [-7.0 1.0203 1.0204 1.0205 1.0206 1.0207 1.0208 1.0209 1.0209 1.021 1.0211]
 [-6.0 1.0194 1.0195 1.0196 1.0197 1.0198 1.0198 1.0199 1.02 1.0201 1.0202]
 [-5.0 1.0185 1.0186 1.0187 1.0187 1.0188 1.0189 1.019 1.0191 1.0192 1.0193]
 [-4.0 1.0176 1.0176 1.0177 1.0178 1.0179 1.018 1.0181 1.0182 1.0183 1.0184]
 [-3.0 1.0166 1.0167 1.0168 1.0169 1.017 1.0171 1.0172 1.0173 1.0174 1.0175]
 [-2.0 1.0157 1.0158 1.0159 1.016 1.0161 1.0162 1.0163 1.0164 1.0165 1.0165]
 [-1.0 1.0148 1.0149 1.015 1.0151 1.0152 1.0153 1.0153 1.0154 1.0155 1.0156]
 [-0.0 1.0139 1.014 1.0141 1.0142 1.0142 1.0143 1.0144 1.0145 1.0146 1.0147]
 [0.0 1.0139 1.0138 1.0137 1.0136 1.0135 1.0134 1.0133 1.0132 1.0131 1.013]
 [1.0 1.013 1.0129 1.0128 1.0127 1.0126 1.0125 1.0124 1.0123 1.0122 1.0121]
 [2.0 1.012 1.0119 1.0118 1.0118 1.0117 1.0116 1.0115 1.0114 1.0113 1.0112]
 [3.0 1.0111 1.011 1.0109 1.0108 1.0107 1.0106 1.0106 1.0105 1.0104 1.0103]
 [4.0 1.0102 1.0101 1.01 1.0099 1.0098 1.0097 1.0096 1.0095 1.0094 1.0094]
```

```
 [5.0 1.0093 1.0092 1.0091 1.009 1.0089 1.0088 1.0087 1.0086 1.0085 1.0084]
 [6.0 1.0083 1.0082 1.0082 1.0081 1.008 1.0079 1.0078 1.0077 1.0076 1.0075]
 [7.0 1.0074 1.0073 1.0072 1.0071 1.007 1.007 1.0069 1.0068 1.0067 1.0066]
 [8.0 1.0065 1.0064 1.0063 1.0062 1.0061 1.006 1.0059 1.0058 1.0057 1.0057]
 [9.0 1.0056 1.0055 1.0054 1.0053 1.0052 1.0051 1.005 1.0049 1.0048 1.0047]
 [10.0 1.0046 1.0045 1.0045 1.0044 1.0043 1.0042 1.0041 1.004 1.0039 1.0038]
 [11.0 1.0037 1.0036 1.0035 1.0034 1.0033 1.0032 1.0032 1.0031 1.003 1.0029]
 [12.0 1.0028 1.0027 1.0026 1.0025 1.0024 1.0023 1.0022 1.0021 1.002 1.0019]
 [13.0 1.0019 1.0018 1.0017 1.0016 1.0015 1.0014 1.0013 1.0012 1.0011 1.001]
 [14.0 1.0009 1.0008 1.0007 1.0007 1.0006 1.0005 1.0004 1.0003 1.0002
  1.0001]
 [15.0 1.0 0.9999 0.9998 0.9997 0.9996 0.9995 0.9994 0.9993 0.9993 0.9992]
 [16.0 0.9991 0.999 0.9989 0.9988 0.9987 0.9986 0.9985 0.9984 0.9983 0.9982]
 [17.0 0.9981 0.998 0.998 0.9979 0.9978 0.9977 0.9976 0.9975 0.9974 0.9973]
 [18.0 0.9972 0.9971 0.997 0.9969 0.9968 0.9967 0.9967 0.9966 0.9965 0.9964]
 [19.0 0.9963 0.9962 0.9961 0.996 0.9959 0.9958 0.9957 0.9956 0.9955 0.9954]
 [20.0 0.9953 0.9953 0.9952 0.9951 0.995 0.9949 0.9948 0.9947 0.9946 0.9945]
 [21.0 0.9944 0.9943 0.9942 0.9941 0.994 0.994 0.9939 0.9938 0.9937 0.9936]
 [22.0 0.9935 0.9934 0.9933 0.9932 0.9931 0.993 0.9929 0.9928 0.9927 0.9926]
 [23.0 0.9926 0.9925 0.9924 0.9923 0.9922 0.9921 0.992 0.9919 0.9918 0.9917]
 [24.0 0.9916 0.9915 0.9914 0.9913 0.9912 0.9912 0.9911 0.991 0.9909 0.9908]
 [25.0 0.9907 0.9906 0.9905 0.9904 0.9903 0.9902 0.9901 0.99 0.9899 0.9898]
 [26.0 0.9898 0.9897 0.9896 0.9895 0.9894 0.9893 0.9892 0.9891 0.989 0.9889]
 [27.0 0.9888 0.9887 0.9886 0.9885 0.9884 0.9883 0.9883 0.9882 0.9881 0.988]
 [28.0 0.9879 0.9878 0.9877 0.9876 0.9875 0.9874 0.9873 0.9872 0.9871 0.987]
 [29.0 0.9869 0.9869 0.9868 0.9867 0.9866 0.9865 0.9864 0.9863 0.9862
  0.9861]
 [30.0 0.986 0.9859 0.9858 0.9857 0.9856 0.9855 0.9854 0.9854 0.9853 0.9852]
 [31.0 0.9851 0.985 0.9849 0.9848 0.9847 0.9846 0.9845 0.9844 0.9843 0.9842]
 [32.0 0.9841 0.984 0.9839 0.9839 0.9838 0.9837 0.9836 0.9835 0.9834 0.9833]
 [33.0 0.9832 0.9831 0.983 0.9829 0.9828 0.9827 0.9826 0.9825 0.9824 0.9824]
 [34.0 0.9823 0.9822 0.9821 0.982 0.9819 0.9818 0.9817 0.9816 0.9815 0.9814]
 [35.0 0.9813 0.9812 0.9811 0.981 0.9809 0.9809 0.9808 0.9807 0.9806 0.9805]
 [36.0 0.9804 0.9803 0.9802 0.9801 0.98 0.9799 0.9798 0.9797 0.9796 0.9795]
 [37.0 0.9794 0.9793 0.9793 0.9792 0.9791 0.979 0.9789 0.9788 0.9787 0.9786]
 [38.0 0.9785 0.9784 0.9783 0.9782 0.9781 0.978 0.9779 0.9778 0.9777 0.9777]
 [39.0 0.9776 0.9775 0.9774 0.9773 0.9772 0.9771 0.977 0.9769 0.9768 0.9767]
 [40.0 0.9766 None None None None None None None None None]]

Various volumes of fuel corrected to 15 degrees Celsius according to the average measured
temperatures of the liquid during the deliveries:
10502.0
10465.0
10295.0
10144.0
10139.0
10133.0
9941.0
9855.0
9766.0
886968.244545
7797325.452114
155.606242
3.95
*************************************************************************************
'''
```