

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی پزشکی

گزارش تمرین پردازش تصویر

تمرین صفر

دانشجو

سیدابوالفضل مرتضوی ۹۸۳۳۰۶۳

مهر ۱۴۰۱

فهرست

تمرین ۱.....	۱
۱-۱.....	۱
۲-۱.....	۱
۳-۱.....	۱
۴-۱.....	۱
۵-۱.....	۲
۶-۱.....	۲
۷-۱.....	۲
تمرین ۲.....	۳
۱-۲.....	۳
۲-۲.....	۳
تمرین ۳.....	۴
۱-۳.....	۴
۲-۳.....	۵
۳-۳.....	۵
تمرین ۴.....	۷
۱-۴.....	۷
۲-۴.....	۷
۳-۴.....	۸
۴-۴.....	۸

تمرین ۱

۱-۱

با استفاده از حلقه for ، ۸۰ عدد داخل لیست ریخته شد و سپس با numpy به آرایه تبدیل شد.

```
[ 617.44 1118.94 163.72 4208.86 -3546.11 3239.27 1797.35 20.48
 -310.89 1561.6 4527.34 4236.62 -4818.98 1927.25 761.3 2329.86
 -2371.75 -2120.67 -4633.04 -471.38 836.39 -1327.88 2970.18 4755.9
 -3956.99 2984.61 3666.46 -4870.09 1812.77 2613.76 -2894.99 -2264.26
 4891.25 -2432.58 2140.44 1103.09 4839.54 -972.63 724.2 1779.69
 4702.97 3660.73 4588.22 2187.43 -2306.78 -1030.15 -83.62 -351.36
 -1666.43 -250.42 -2359.49 -68.49 -2740.15 -1577.12 2468.21 654.94
 657.32 2793.63 -533.92 -4348.35 -3103.35 304.9 2463.11 4569.68
 3687.02 -1610.33 1538.39 -452.49 2412.27 -781.92 -3194.56 170.51
 -717.39 -4621.94 1446.11 3426.17 1304.82 4413.81 1906.34 3747.43]
```

۲-۱

```
<class 'numpy.ndarray'>
float64
```

۳-۱

با استفاده از دستور round این کار انجام شد.

```
[-2985. -819. 3746. -3637. -3600. 361. -3577. 4850. 1741. -2740.
 1485. -2786. 1842. -4509. -525. 2769. -2418. -3825. -4803. -1430.
 4222. 1988. -3684. -1716. 269. 4222. -4779. -4055. 3321. -653.
 -3859. -4440. 1280. 4308. -2850. -582. 4137. -3912. -2119. -4838.
 -2380. 3291. 2956. -4549. -2082. 1502. -4457. -4434. -3696. 4598.
 1881. 478. -820. -3066. 579. -2995. 994. -1133. -3214. 903.
 1081. -760. -3377. -3263. 4783. -849. -3942. 3863. -1076. -957.
 3429. -412. -2283. -4067. 4544. 636. -3211. 579. 3653. 4160.]
```

۴-۱

برای تبدیل نوع از int16 استفاده شد.

۵-۱

با استفاده از روابط ریاضی اعداد به بازه موردنظر انتقال یافتند.

```
Random_array=(Random_array+5000)/39.21
Random_array=Random_array.astype('float32')
print(Random_array)
```

```
[227  90 203  17  11  25 140 112  24 221 210 126  71  54 146  89 198 228
122  52  50 115 127  23  37  45 142 154  61 170 113 174 184   3 103  56
153 143 118 139  50  74   7  31  76 251 206   4 138 243  87 240  92 175
115 184  37 100 138  14  31  69 105 146 136  63  49 105 119  56 204 246
205 164  55 107  62 137  67 189]
```

۶-۱

با دستور reshape آرایه را تبدیل کردم

```
[[242  52  88  35  14 189  92  51]
 [175  87 157  82 132  23 196 246]
 [224  96 244  99 231  46  24   8]
 [190 195  94 253 111 253 110 136]
 [193  18  59  27  80 117  33 208]
 [175  84 227  56  10 215 209  73]
 [220  88 177 148 159 170 131  34]
 [222 229  64 205  99  65 232 186]
 [242  30 166  39 175  63 167   6]
 [148 211 140   4 105 242  11 107]]
```

۷-۱

با توجه به اینکه این نوع داده بازه ۲۵۵،۲۵۵- را شامل می شود اعداد را با جمع و... به این بازه انتقال می دهد

و اعداد با اعداد اصلی متفاوت می شوند.

```
[[ -27  -83   43  -61   97  -38 -100   69]
 [-124   71  -26  118   46   64   84  -59]
 [ -71  -10  -98  -39   59   58 -119 -105]
 [ -30  -29   24  105   95 -108  -10 -109]
 [ -67  121   21  -33  -84   -9   65  -24]
 [ -69   77   90 -108   13   67  -70  -19]
 [  40 -128  -21  114 -103   38   55  -25]
 [-125   87 -124   91  -35  -35   -4   47]
 [   5  127   20  -82  -83  -92  -52  -38]
 [-11   82   53  -43  -72  -70  -76  -51]]
```

تمرین ۲

۱-۲

تابع مورد نظر تعریف شد و با دستور @typechecked نوع ورودی آن محدود شد تا به جز نوع های گفته شده نوع دیگری را قبول نکند

```
from typeguard import typechecked
@typechecked #limit input type
def matrix(seed:int,dims:tuple):
```

۲-۲

تابع به صورت زیر تعریف شد و خروجی آن هم در تصاویر بعدی آورده شده است.

```
def matrix(seed:int,dims:tuple):
    (rows, cols) =dims
    matrix2D = [[0]*cols for i in range(rows)]
    for i in range(rows):
        for j in range(cols):
            if i==0 and j==0:
                matrix2D[i][j]=seed
            elif i==0:
                matrix2D[i][j]=matrix2D[i][j-1]
            elif j==0:
                matrix2D[i][j]=matrix2D[i-1][j]
            else:
                matrix2D[i][j]=matrix2D[i-1][j]+matrix2D[i][j-1]+matrix2D[i-1][j-1]
    matrix2D=np.array(matrix2D)
    matrix2D=matrix2D.reshape(rows,cols)
    return matrix2D
```

```
[ [ 2  2  2  2]
  [ 2  6 10 14]
  [ 2 10 26 50]]
```

تمرین ۳

۱-۳

```
import numpy as np
from matplotlib import pyplot as pp
from math import floor
def unit_circle_vectorized(r):
    A = np.arange(-r,r+1)**2
    dists = np.sqrt(A[:,None] + A)
    return 255*((dists-r)<(0.5)**r).astype('uint8')
print(unit_circle_vectorized(7))
print(type(unit_circle_vectorized(7)))
```

این تابع به شکل بالا نوشته شد و خروجی های آن برای اعداد ۳ و ۵ به شکل زیر است.

```
[ [ 0  0  0  0  0 255  0  0  0  0  0]
  [ 0  0 255 255 255 255 255 255 255  0  0]
  [ 0 255 255 255 255 255 255 255 255 255  0]
  [ 0 255 255 255 255 255 255 255 255 255  0]
  [255 255 255 255 255 255 255 255 255 255 255]
  [ 0 255 255 255 255 255 255 255 255 255  0]
  [ 0 255 255 255 255 255 255 255 255 255  0]
  [ 0 255 255 255 255 255 255 255 255  0  0]
  [ 0  0 255 255 255 255 255 255 255  0  0]
  [ 0  0  0  0  0 255  0  0  0  0  0]]
```

```
[ [ 0  0  0 255  0  0  0]
  [ 0 255 255 255 255 255  0]
  [ 0 255 255 255 255 255  0]
  [255 255 255 255 255 255 255]
  [ 0 255 255 255 255 255  0]
  [ 0 255 255 255 255 255  0]
  [ 0  0  0 255  0  0  0]]
```

۲-۳

```
#3b
stdnumber=9833063
def noise_circle(Arr):
    (row,col)=np.shape(Arr)
    for i in range(row):
        for j in range(col):
            if Arr[i][j]==0:
                Arr[i][j]+=np.random.uniform(0,48)
                Arr[i][j]=floor(Arr[i][j])
            else:
                Arr[i][j]-=np.random.uniform(0,48)
                Arr[i][j]=floor(Arr[i][j])
    return Arr
```

این تابع به شکل بالا نوشته شد و خروجی های آن برای اعداد ۳ و ۵ به شکل زیر است.

```
[[ 47  7 25 21 35 211 43 35 45 11 46]
 [ 25  0 234 240 235 215 235 226 214 14 9]
 [ 43 225 239 231 234 238 240 253 231 249 1]
 [ 24 210 245 245 215 235 245 240 220 248 27]
 [ 45 238 227 217 250 237 219 208 242 236 39]
 [251 216 222 238 231 251 250 229 216 210 216]
 [ 4 208 234 244 221 239 220 218 239 239 32]
 [ 20 246 240 234 207 218 217 221 241 215 0]
 [ 35 209 217 230 218 220 229 228 213 248 41]
 [ 14 40 217 228 207 217 227 223 226 24 7]
 [ 10 35 16 26 38 249 25 3 35 12 8]]
```

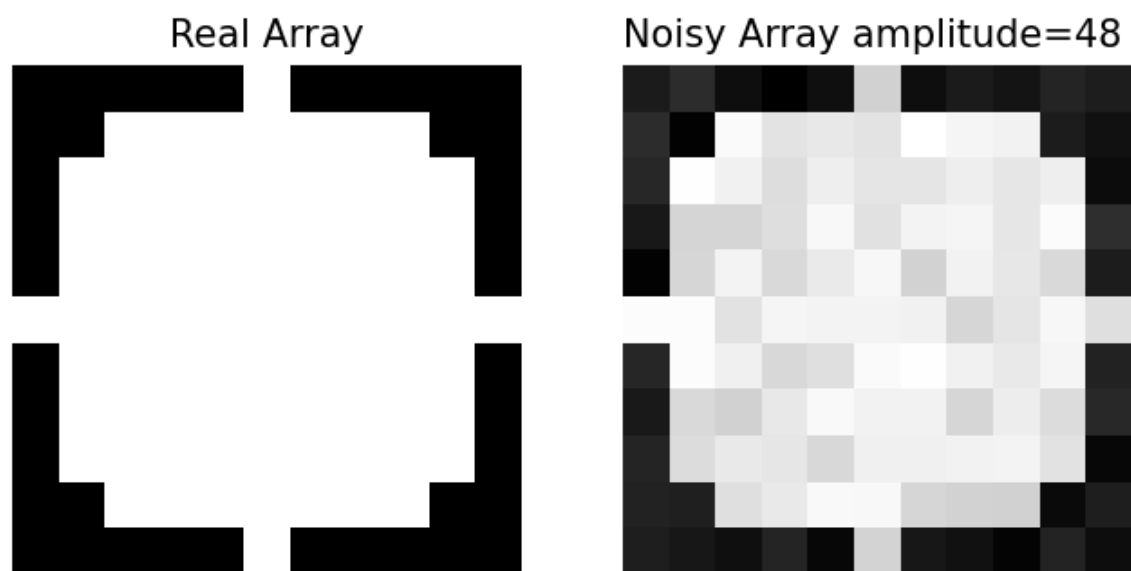
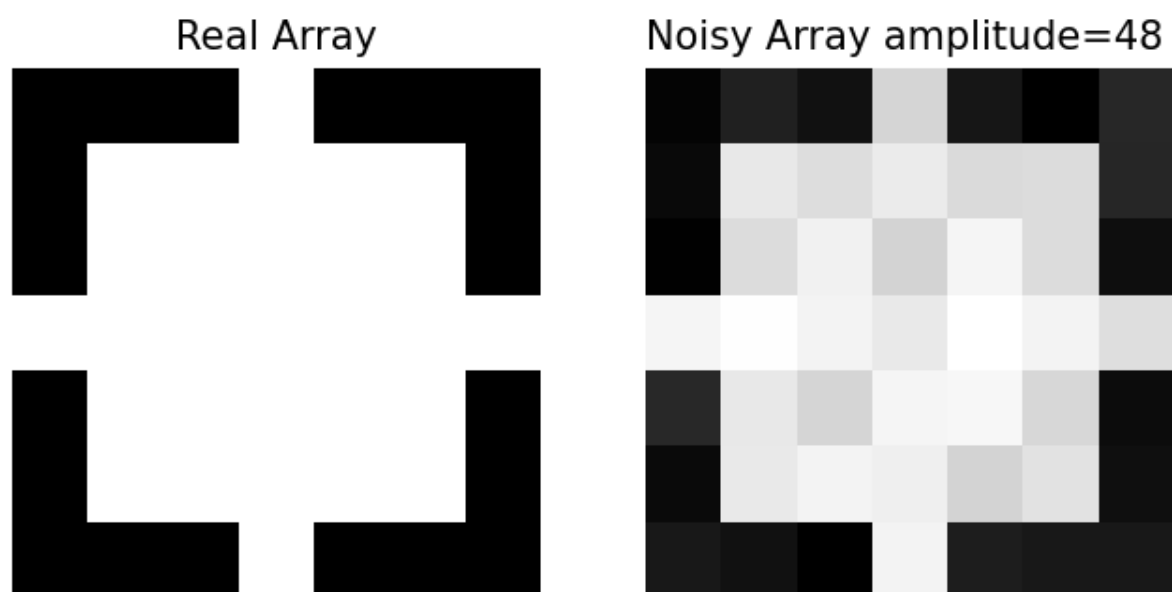
```
[[ 15  6 36 210 46 14 13]
 [ 9 208 213 228 228 233 30]
 [ 19 210 245 225 216 237 36]
 [218 235 250 220 242 210 227]
 [ 33 214 229 252 221 240 17]
 [ 19 247 243 222 236 214 1]
 [ 35 15 44 228 30 5 3]]
```

۳-۳

کد نوشته شده به صورت زیر است.

```
pp.figure('HW0-Image-9833063')
pp.subplot(1,2,1)
pp.title('Real Array')
pp.imshow(unit_circle_vectorized(3),'gray')
pp.axis(False)
pp.subplot(1,2,2)
pp.title('Noisy Array amplitude=48 ')
pp.imshow(noise_circle(unit_circle_vectorized(3)),'gray')
pp.axis(False)
pp.savefig('HW0-9833063-3.png')
pp.show()
```

تصاویر خروجی برای شعاع ۵۳ به صورت زیر است.



تمرین ۴

۱-۴

کد نوشته شده به صورت زیر است.

```
#4a
chest_im=cv.imread('ChestXray.tif')
cv.imshow('Display window',chest_im)
print(np.shape(chest_im))
chest_im_gray=cv.cvtColor(chest_im,cv.COLOR_BGR2GRAY)
# cv.imshow('Display window',chest_im_gray)
print(np.shape(chest_im_gray))
```

تصویر خروجی به شکل زیر است.



ابعاد تصویر جدید و بدون تغییر در رنگ به شکل زیر است.

```
(493, 600, 3)
(493, 600)
```

۲-۴

نوع داده برای هرپیکسل در هر دو شکل uint8 است.

```
uint8
uint8
```

۳-۴

در شکل زیر این اعداد آمده است. عدد اول مربوط به شکل خاکستری و دیگری مربوط به شکل رنگی است.

یکای داده ها بیت است.

295800
887400

۴-۴

کد نوشته شده برای قسمت های آخر سوال به صورت زیر است.

```
#4d
crop_img=chest_im_gray[80:, 100:500]
cv.imshow("cropped", crop_img)
#
#4e
mirror=chest_im_gray[80:, 500:1000:-1]
cv.imshow("mirror", mirror)
#4H
plt.figure("HW0-9833063")
plt.subplot(1,3,1)
plt.title('Gray Image')
plt.imshow(chest_im_gray, 'gray')
plt.axis(False)
plt.subplot(1,3,2)
plt.title('Cropped Image')
plt.imshow(crop_img, 'gray')
plt.axis(False)
plt.subplot(1,3,3)
plt.title('Mirrored Image')
plt.imshow(mirror, 'gray')
plt.axis(False)
plt.savefig('HW0-9833063-4.png')
plt.show()
cv.waitKey(0)
```

خروجی تصاویر بریده شده و قرینه شده و تصویر اصلی به صورت زیر است.

