

به نام خدا

علی انصاری

۹۸۳۱۱۳۶

۱.

۱. با استفاده از دستور :

```
arr = np.random.uniform(-5000, 5000, size=80)
```

۸۰ عدد تصادفی را میگیریم.

```
Step 1
=====
[-3923.5487775 -2068.94043973 3244.49603958 -3453.80033647
 2166.39560503 4619.14053524 -2563.91034529 3405.55152491
 2338.87650198 -816.192571 -3397.99756545 -1071.0072446
-1781.014367 3354.45822817 -905.30469021 -3553.30808169
-1317.30166419 1911.90742122 3101.74620136 -2010.95540484
-4988.20270894 3447.84313985 -2653.14676808 -100.34210589
-2096.75011794 3289.64766621 174.39650844 -3730.18155894
-3574.01422759 -4655.93683201 -1213.37127717 4510.33216302
-526.67314536 -1081.31650986 -3462.0938517 -82.71615983
-2492.55788047 212.94390456 1721.57900961 2174.18839942
 4278.35831735 3259.58053588 -533.63325663 4040.54629924
-1483.48605525 3970.77697716 1284.13121865 3928.45587545
 3213.06824332 -4672.82779626 1745.06916932 1563.63539483
-2718.22313323 -397.63183051 3116.87634396 3557.18775488
 1481.24969306 -910.03501922 -1972.39640782 4684.05871859
-487.35080296 -212.01853988 -3956.38444859 -1717.14428218
-4039.89390422 71.6733248 -1293.31143195 4846.66797634
-4356.33429282 -2789.83692031 -2776.72816887 4023.37502202
-1717.81328547 -4629.86504673 -2804.16748991 4285.00230883
-165.51294994 -1116.52941053 1001.60456802 -4714.41080683]
```

۲. با استفاده از دو دستور:

```
print(arr.dtype, type(arr))
```

به ترتیب نوع داده آرایه و همچنین نوع خود آرایه را میگیریم که به ترتیب float64 و ndarray هستند.

```
Step 2
=====
float64 <class 'numpy.ndarray'>
```

۳. با استفاده از دستور:

```
arr = [round(i) for i in arr]
```

اعداد را به سمت نزدیکترین عدد صحیح گرد میکنیم.

```
Step 3
=====
[-3924. -2069. 3244. -3454. 2166. 4619. -2564. 3406. 2339. -816.
-3398. -1071. -1781. 3354. -905. -3553. -1317. 1912. 3102. -2011.
-4988. 3448. -2653. -100. -2097. 3290. 174. -3730. -3574. -4656.
-1213. 4510. -527. -1081. -3462. -83. -2493. 213. 1722. 2174.
 4278. 3260. -534. 4041. -1483. 3971. 1284. 3928. 3213. -4673.
 1745. 1564. -2718. -398. 3117. 3557. 1481. -910. -1972. 4684.
-487. -212. -3956. -1717. -4040. 72. -1293. 4847. -4356. -2790.
-2777. 4023. -1718. -4630. -2804. 4285. -166. -1117. 1002. -4714.] float64
```

۴. نوع متغیر int16 مناسب برای این قسمت است زیرا از بازه اعداد ما بزرگتر است.

```
Step 4
=====
[ -3924 -2069  3244 -3454  2166  4619 -2564  3406  2339  -816 -3398 -1071
 -1781  3354  -905 -3553 -1317  1912  3102 -2011 -4988  3448 -2653 -100
 -2097  3290  174  -3730 -3574 -4656 -1213  4510  -527 -1081 -3462  -83
 -2493  213  1722  2174  4278  3260  -534  4041 -1483  3971  1284  3928
  3213 -4673  1745  1564 -2718  -398  3117  3557  1481  -910 -1972  4684
 -487  -212 -3956 -1717 -4040   72 -1293  4847 -4356 -2790 -2777  4023
 -1718 -4630 -2804  4285  -166 -1117  1002 -4714] int16
```

۵. با استفاده از دستور:

```
arr = np.interp(arr, (-5000, 5000), (0, 255))
```

آرایه را به بازه ۰ تا ۲۵۵ انتقال میدهیم.

```
Step 5
=====
[ 27  74 210  39 182 245  62 214 187 106  40 100  82 213 104  36  93 176
 206  76   0 215  59 124  74 211 131  32  36   8  96 242 114  99  39 125
  63 132 171 182 236 210 113 230  89 228 160 227 209   8 171 167  58 117
 206 218 165 104  77 246 115 122  26  83  24 129  94 251  16  56  56 230
  83   9  55 236 123  99 153   7] int16
```

۶. با دستور arr.shape(8,10) ابعاد آرایه را تغییر میدهیم.

```
Step 6
=====
[[ 27  74 210  39 182 245  62 214 187 106]
 [ 40 100  82 213 104  36  93 176 206  76]
 [   0 215  59 124  74 211 131  32  36   8]
 [ 96 242 114  99  39 125  63 132 171 182]
 [236 210 113 230  89 228 160 227 209   8]
 [171 167  58 117 206 218 165 104  77 246]
 [115 122  26  83  24 129  94 251  16  56]
 [ 56 230  83   9  55 236 123  99 153   7]]
```

۷. با دستور arr.astype(np.uint8) فرمت داده‌های داخل آرایه را تغییر میدهیم که بازه آن از ۱۲۸- تا ۱۲۷ است. اعداد بزرگتر از ۱۲۷ تغییر میکنند زیرا از بازه ۰ تا ۲۵۵ به بازه ۱۲۸- تا ۱۲۷ تبدیل میشوند.

```
Step 7
=====
[[ 27  74 -46  39 -74 -11  62 -42 -69 106]
 [ 40 100  82 -43 104  36  93 -80 -50  76]
 [   0 -41  59 124  74 -45 -125  32  36   8]
 [ 96 -14 114  99  39 125  63 -124 -85 -74]
 [-20 -46 113 -26  89 -28 -96 -29 -47   8]
 [-85 -89  58 117 -50 -38 -91 104  77 -10]
 [115 122  26  83  24 -127  94  -5  16  56]
 [ 56 -26  83   9  55 -20 123  99 -103   7]]
```

۲. با توجه به توضیح سوال فکر میکنم $seed=2$ باید باشد در مثال آورده شده. تصویر خروجی مانند مثال را در زیر آورده‌ام.

```
PS C:\Users\MY PC> & "C:\Program Files\Python37\python.exe" & "C:\Users\MY PC\Desktop\2.py"
2 (3, 4)

2 2 2 2
2 6 10 14
2 10 26 50
```

مثال دیگری نیز میتوانیم نمایش دهیم.

```
PS C:\Users\MY PC\Desktop> "C:\Program Files\Python37\python.exe" & "C:\Users\MY PC\Desktop\3.py"
3 (5, 6)

3 3 3 3 3 3
3 9 15 21 27 33
3 15 39 75 123 183
3 21 75 189 387 693
3 27 123 387 963 2043
```

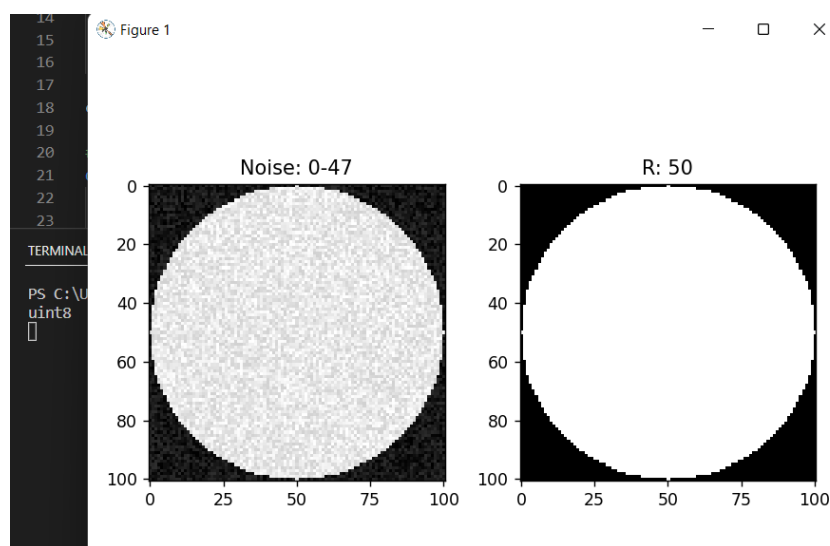
۳. آرایه را با دستور `astype(np.uint8)` به نوع گفته شده تغییر دادیم. همچنین در نهایت که تمامی اعمال ها را روی آرایه انجام دادیم دوباره چک کردیم که نوع داده ما تغییر نکرده باشد که در خروجی مشاهده میشود. دایره را با فرمول :

```
if (i-r)**2 + (j-r)**2 <= r**2:
    arr[i,j] =255
```

رسم کردیم.

سپس نویز را نیز با توجه به فرمول گفته شده به دایره اضافه کردیم.

تصویر خروجی را در فایل فرستاده شده نیز با فرمت ذخیره گفته شده پیوست کرده‌ام.



۱. به دلیل اینکه مقادیر هر سه آرایه رنگی برابر است، هرکدام از رنگ‌ها را بگیریم تفاوتی ندارد و تصویر به حالت خاکستری میشود. ابعاد تصویر برابر (493,600) است و قبل از تبدیل به حالت خاکستری ۴ بعد دارد اما پس از آن تنها یک بعد که ۰ تا ۲۵۵ است دارد.

۲. نوع داده‌های پیکسل uint8 است.

۳. حافظه اشغال شده را با دستور `getsizeof(img)` به دست می‌آوریم که برای تصویر رنگی برابر با ۱۱۸۳۳۳۶ و برای خاکستری برابر با ۱۲۰ بایت بود.

۴ و ۵ و ۶. برای قسمت شش `vmin = 0` و `vmax = 255` قرار دادیم. تصویر تمامی خروجی‌های شش قسمت را در قسمت زیر آورده ام.

