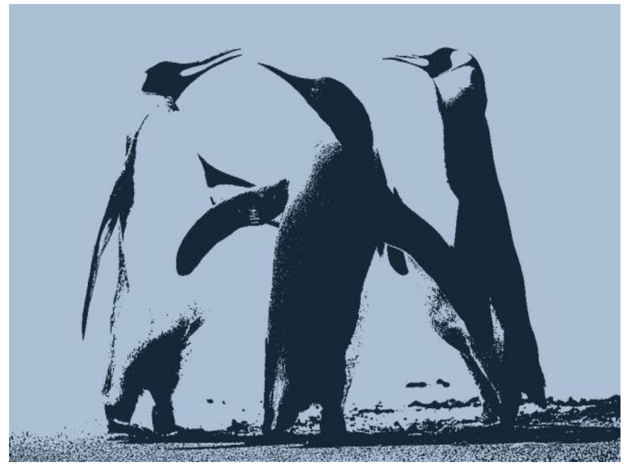
1. The images after data compression using different values of K.
    - For penguins:

Original Image:

K=2:



K=5:

K=10:

K=15:

K=20:

- For Koala:

Original Image:

K=2:

K=5:

K=10:

K=15:

K=20:

2. Original file size of Koala.jpg is 762KB, and Penguin.jpg is 759KB
   I ran each K value for 5 times with different random initialization, and only consider precision to KB.
   Koala.jpg:

K=2:	127KB, 127KB, 127KB, 127KB, 127KB	Average:127KB	Variance:0
K=5:	171KB, 172KB, 172KB, 172KB, 172KB	Avarege:171.6KB	Variance:0.24
K=10:	161KB, 159KB, 161KB, 161KB, 160KB	Avarege:160.4KB	Variance:0.64
K=15:	153KB, 155KB, 155KB, 154KB, 155KB	Avarege:154.4KB	Variance:0.64
K=20:	152KB, 152KB, 153KB, 151KB, 152KB	Average:152KB	Variance:0.4

   Compression ratio for Koala.jpg:

K=2:	6
K=5:	4.44
K=10:	4.75
K=15:	4.94
K=20:	5.01

   Penguins.jpg:

K=2:	83KB, 83KB, 83KB, 83KB, 83KB	Average: 83KB	Variance:0
K=5:	102KB, 104KB, 103KB, 102KB, 103KB	Avarege:102.8KB	Variance:0.56
K=10:	114KB, 114KB, 111KB, 114KB, 116KB	Avarege:113.8KB	Variance:2.56
K=15:	112KB, 110KB, 114KB, 114KB, 111KB	Avarege:112.2KB	Variance:2.56
K=20:	113KB, 112KB, 113KB, 113KB, 112KB	Average:112.2KB	Variance:0.23

   Compression ratio for Penguins.jpg

K=2:	9.14
K=5:	7.38
K=10:	6.67
K=15:	6.76
K=20:	6.76

   I noticed that the compression ration does not change much after K>5. This is because in the template, we output the image as a jpeg file. Jpeg has color depth of 24bits, which means even though we reduce the total 2^24 colors to logK color palette, we still need 24bits to represent the rgb value for each pixel.
   Ideally, if we can customize our own color model, and reduce the color depth, which uses less bits to represent the color index in each pixel, and mapping these indexes to the 24bits color palette, we can greatly reduce the size of the output image file size.

   Theoretically, the original image take up w*h*24bits, where w width and h is height, since each pixel needs 24bits to represent rgb value for color. However, after compression, if we reduce the color depth, we can use w*h*logK to represent color indexes, and K*24bits, therefore (w*h*logK + k*24)bits in total.
   The theoretical compression ratio will be (w*h*24)/(w*h*logK + k*24).

3.
There is a tradeoff when K<5. The compression ratio decreases as the image quality gets better. However, after K>5, the image quality gets better while the compression ration stay the same. I would choose a greater K because the quality is better. However, it takes longer to compute, and our eyes cannot tell the difference when K is a lot larger, so I think K=20 is good enough for two images.