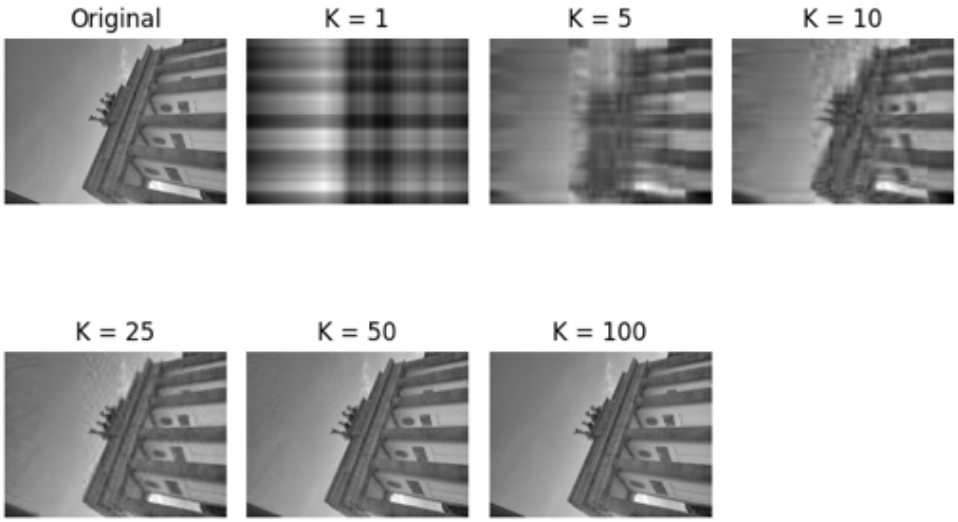
In [ ]:
```
!pip install numpy
!pip install matplotlib
!pip install imageio
```

In [41]:
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
```

In [ ]:

```
class H07:
    def __init__(self,img_path):
        self.img_path = img_path # das gegebene Bild in python
einlesen
    # das eingelesene Bild in ein schwarz-weiß Bild
konvertieren
    def img_2_gray(self):
        img_color = Image.open(self.img_path)
        # "L" = 8-bit grayscale
        img_gray  = img_color.convert("L")
        # shape (m, n)
        A = np.array(img_gray, dtype=float)
        return A
    # use the function as the task specified
    def reconstruct_svd(self,U, s, Vh, K):
        Uk  = U[:, :K]
        sk  = s[:K]
        Vhk = Vh[:K, :]
        return Uk @ np.diag(sk) @ Vhk
    # K ∈{1,5,10,25,50,100}, it is given in the task
description
    def get_Ks(self):
        Ks = [1, 5, 10, 25, 50, 100]
        return Ks
    def plot_result(self,A,Ks,U,s,Vh):
        plt.figure(figsize=(7, 5))
        plt.subplot(2, 4, 1)
        plt.title("Original")
        plt.imshow(A, cmap="gray")
        plt.axis("off")
        for i, K in enumerate(Ks, start=2):
            A_K = self.reconstruct_svd(U, s, Vh, K)
            A_K = np.clip(A_K, 0, 255)
            plt.subplot(2, 4, i)
            plt.title(f"K = {K}")
            plt.imshow(A_K, cmap="gray")
            plt.axis("off")
        plt.tight_layout()
        plt.show()
    # apply the functions on the image given by the task
    def apply_h07_on_img(self):
        A = self.img_2_gray()
        # benutzen Sie die svd Funktion in numpy.linalg um das
Bild zu zerlegen.
        U, s, Vh = np.linalg.svd(A, full_matrices=False)
        Ks = self.get_Ks()
        self.plot_result(A,Ks,U,s,Vh)
```

In [43]:
```
gate = H07("gate.jpeg")
gate.apply_h07_on_img()
```



In [40]:
```
hall = H07("concert_hall.jpeg")
hall.apply_h07_on_img()
```