25 Gaussian Blur

13 points

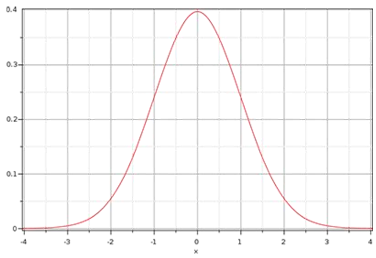
Introduction

For sure, you've seen many times in photos, the typical effects where the out-of-focus parts are softened or even the whole image seems like viewed through a translucent screen.

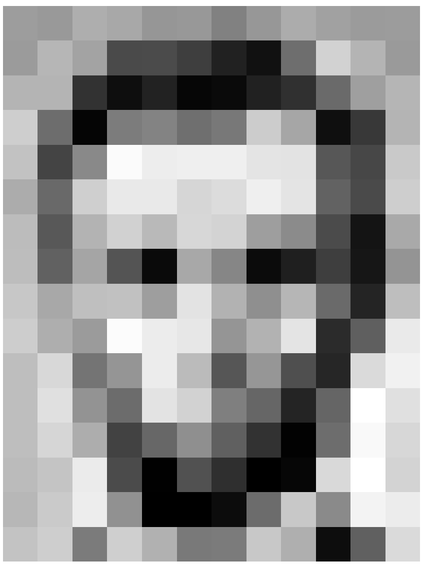
Original image

After applying Gaussian Blur

Some of these effects are accomplished using a Gaussian Blur which is the application of a mathematical function to an image in order to blur it. A Gaussian curve or Gaussian function has the following shape:



As you may know an image is represented as a matrix of pixel values.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

So let's consider a simple way to apply a Gaussian Blur to a given image. To make things easier assume that only some points along the Gaussian curve are considered. These points, also called "weights", show how much softness we want at that place. Higher numbers mean more softness. The selected points are collected in a one-dimensional weight table like this:

```
[0.06136 0.24477 0.38774 0.24477 0.06136]
```

This is also called Gauss kernel window. Then, placing this window centered on a certain pixel, we would sample every other pixel in the window using the corresponding weight, and the sum of all them would be the blurred value of the pixel in the center:

Image size = 7 x 7

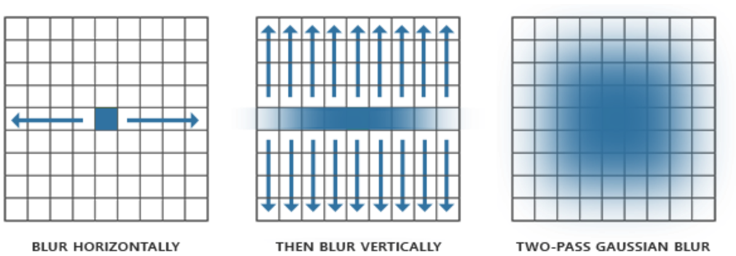
125	125	75	125	255	75	45
100	100	75	50	40	25	0
100	150	175	190	225	245	255
255	255	250	245	255	255	255
250	245	240	235	225	220	215
200	150	200	100	150	100	50
50	25	20	0	25	25	0

Gauss kernel window

0.06136	0.24477	0.38774	0.24477	0.06136

Original Pixel Value = 75
Blurred Pixel Value = 100 x 0.06136 + 100 x 0.24477 + 75 x 0.38774 + 50 x 0.24477 + 40 x 0.06136 = 74.386

The operation to be done consists of a first horizontal blur on the image, followed by another vertical blur on the resulting image:



So, for every pixel we'll put the one-dimensional window horizontally around it and compute the blurred value using the weights in the window. Then, in the resulting image, for each pixel we'll put the one-dimensional window vertically around it and compute the blurred value using the weights. We can repeat this process any number of times if we want a more blurred result. Corner pixels and pixels situated along the sides of the image are missing some neighboring pixels. When placing the window around these pixels, assume that the value of missing neighbors is '0'.

Write a program that applies the Gaussian Blur a specified number of times to a given image using previous one-dimensional window weight table.

Input

First line, a positive number indicating how many times we want to blur the image. Then, a line with the size of the image separated by a white space (rows columns).

And finally, the image pixel values.

Output

The output is the pixel values of the blurred image, rounded to the nearest integer.

Example

Input

```
4
5 5
73 54 80 45 73
37 63 54 54 18
97 37 80 80 97
27 45 97 18 18
45 45 18 45 97
```

Output

```
10 16 18 16 10
16 25 28 25 16
18 28 32 28 18
15 24 27 24 15
9 15 17 15 10
```