Grayscale

Rar the Cat has 2 images that are both **H** pixels by **W** pixels. Each pixel of the image is a colour represented in the RGB colour model. The RGB colour model is an additive model in which red, green and blue lights are added together in various compositions to reproduce a broad array of colours. Just like in most implementations, each of the three colours (red, green and blue) can be a value from 0 to 255, depending on the intensity of the respective colours.

As an example, the following are different colours represented as values in the RGB model:

Colour	R	G	B
Black	0	0	0
Red	255	0	0
Green	0	255	0
Blue	0	0	255
White	255	255	255

Rar the Cat wants to compare the images in *grayscale*. In grayscale, pixels are only expressed as an intensity of black/white, instead of colours. Each pixel in grayscale will take a value of 0 to 255. When the pixel has a value of 0, it would have the darkest shade of black, and 255 will represent the colour of white. Values in between 0 and 255 would just be different intensities of gray, with lower values representing shades of gray that are darker than higher values.

To compute the grayscale value of a pixel from the RGB pixel values, Rar the Cat proposes to take the **average of all RGB pixel values that are within a 3 by 3 square centred at the pixel, rounded down.** For example, consider the below 3 by 3 pixel images:

Red

255	255	224
255	255	249
243	244	242

Green

255	255	254
255	255	245
249	253	247

Blue

255	255	253
255	255	250
250	244	251

=

Grayscale

255	251	250
252	250	249
251	249	249

In the grayscale image above, the centre cell is calculated as the average of all the RGB pixel cells in the whole 3 by 3 image. This is because all the cells are within the 3 by 3 square that is centred at the centre cell.

Red

255	255	224
255	255	249
243	244	242

Green

255	255	254
255	255	245
249	253	247

Blue

255	255	253
255	255	250
250	244	251

=

Grayscale

255	251	250
252	250	249
251	249	249

However, the bottom left pixel (*251*) is calculated as { (255 + 255 + 243 + 244) + (255 + 255 + 249 + 253) + (255 + 255 + 250 + 244) } / 12, rounded down. This corresponds to the average of the shaded cells in the diagram above. The other cells are not included in the average as they are not within the 3 by 3 square which is centred at the bottom left pixel.

The page will depict a few more examples on which cells are used to calculate the grayscale value for a 3 by 3 image.

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

Red			Green			Blue				Grayscale		
255	255	224	255	255	254	255	255	253	=	255	251	250
255	255	249	255	255	245	255	255	250		252	250	249
243	244	242	249	253	247	250	244	251		251	249	249

The page will depict a few more examples on which cells are used to calculate the grayscale value for a 4 by 4 image.

Red

52	57	29	15
86	49	22	10
89	44	0	1
99	92	2	0

Green

22	27	29	18
26	29	22	17
19	14	20	24
29	21	24	23

Blue

0	2	5	9
3	34	50	92
4	24	51	97
7	17	68	98

=

Grayscale

32	30	28	26
32	29	29	28
38	35	35	34
38	34	34	34

Red

52	57	29	15
86	49	22	10
89	44	0	1
99	92	2	0

Green

22	27	29	18
26	29	22	17
19	14	20	24
29	21	24	23

Blue

0	2	5	9
3	34	50	92
4	24	51	97
7	17	68	98

=

Grayscale

32	30	28	26
32	29	29	28
38	35	35	34
38	34	34	34

Red

52	57	29	15
86	49	22	10
89	44	0	1
99	92	2	0

Green

22	27	29	18
26	29	22	17
19	14	20	24
29	21	24	23

Blue

0	2	5	9
3	34	50	92
4	24	51	97
7	17	68	98

=

Grayscale

32	30	28	26
32	29	29	28
38	35	35	34
38	34	34	34

Red

52	57	29	15
86	49	22	10
89	44	0	1
99	92	2	0

Green

22	27	29	18
26	29	22	17
19	14	20	24
29	21	24	23

Blue

0	2	5	9
3	34	50	92
4	24	51	97
7	17	68	98

=

Grayscale

32	30	28	26
32	29	29	28
38	35	35	34
38	34	34	34

Red

52	57	29	15
86	49	22	10
89	44	0	1
99	92	2	0

Green

22	27	29	18
26	29	22	17
19	14	20	24
29	21	24	23

Blue

0	2	5	9
3	34	50	92
4	24	51	97
7	17	68	98

=

Grayscale

32	30	28	26
32	29	29	28
38	35	35	34
38	34	34	34

Red

52	57	29	15
86	49	22	10
89	44	0	1
99	92	2	0

Green

22	27	29	18
26	29	22	17
19	14	20	24
29	21	24	23

Blue

0	2	5	9
3	34	50	92
4	24	51	97
7	17	68	98

=

Grayscale

32	30	28	26
32	29	29	28
38	35	35	34
38	34	34	34

Before comparison, Rar the Cat wants you to **convert both images to grayscale**. In order to account for differences in exposure, Rar the Cat has downloaded a photo editing software. This photo editing software allows Rar the Cat to increase or decrease *all* pixel values in the image by an integer value. This integer can be **negative integer, positive integer or even 0**. However, if the **pixel value goes below 0, it would become 0**. If the **pixel value goes above 255, it will remain at 255**. For simplicity, we will call this a *transpose operation*.

After transposing, Rar the Cat will then compare the similarity of the two images (now grayscale) by calculating the *sum of differences*. The *sum of differences* is calculated by taking the sum of absolute differences in pixel values between the pixels in the same position of the two images. The diagram on the following page details an example.

In the diagram below, we have two 3 by 3 grayscale images. The absolute differences in pixel values are illustrated in the 3 by 3 matrix on the right. The *sum of differences* is hence the sum of the matrix on the right, which adds up to **16**.

Image 1
(Grayscale)

255	251	250
252	250	249
251	249	249

Image 2
(Grayscale)

254	253	253
253	252	251
253	251	250

Absolute
Difference

1	2	3
1	2	2
2	2	1

Unexpectedly, the photo editing software that Rar the Cat has downloaded is a *trial version* that restricts the user to perform only one *transpose operation*. As such, Rar the Cat can only perform one transpose operation on either image 1 or image 2. Out of all the possible transpose operations that Rar the Cat can perform, he is interested to know the **minimum *sum of differences* that the two images can have**, after just **one *transpose operation***.

Input

The first line of input will contain 2 space-separated integers, **H**, followed by **W**. This denotes that the two images are **H** pixels high and **W** pixels wide. In addition, it is guaranteed that **0 < H ≤ 50** and **0 < W ≤ 50.**

6 *blocks* will follow. In each block, there will be **H** lines containing **W** space-separated integers each. The j^{th} integer in the i^{th} row will describe the pixel that is in the i^{th} row, j^{th} column (1-indexed). It is guaranteed that all integers in each block will be between 0 and 255.

Each image will be described in 3 *blocks* as defined above. The first three blocks will describe the red, green and blue pixel values of the first image, respectively. The next three blocks will describe the red, green and blue pixel values of the second image, respectively. In addition, there will be an empty line right before every block (see Sample Input).

Output

Print a single integer representing the minimum *sum of differences* the two images can have in grayscale, after one transpose operation has been performed.

Sample Input

```
3 3

255 255 224
255 255 249
243 244 242

255 255 254
255 255 245
249 253 247

255 255 253
255 255 250
250 244 251

255 255 251
255 255 244
253 254 234

255 255 254
253 255 255
250 247 250

255 255 253
255 255 252
253 254 251
```

Sample Output

```
4
```

Explanation

The grayscale values of the two images are as computed in the examples.

Image 1 (Grayscale)

255	251	250
252	250	249
251	249	249

Image 2 (Grayscale)

254	253	253
253	252	251
253	251	250

When we transpose image 1 by a value of **+2**, we obtain the following result and the *sum of differences* is **4**. This is the minimum possible *sum of differences* for all possible transpose operations.

Image 1 (Transposed)

255	253	252
254	252	251
253	251	251

Image 2

254	253	253
253	252	251
253	251	250

Difference

1	0	1
1	0	0
0	0	1

Notice that for the top left pixel, its value remains at 255 even after transposing.

Skeleton

You are given the skeleton file *Grayscale.java*. You should see a non-empty file when opening it, otherwise you are in the wrong directory.

Notes:

1. You should develop your program in the subdirectory `ex1` and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. If your algorithm is different from the given skeleton, you are free to write a solution according to your own algorithm.
3. You do not need to use OOP for this sit-in lab.
4. You are free to define your own helper methods. **Remember to use private methods whenever possible.**
5. Please be reminded that the marking scheme is:

   Input : 10%
   Output : 10%
   Correctness : 50%
   Programming Style : 30%, which consists of:
   - Meaningful comments (pre- and post- conditions, comments inside the code): 10%
   - Modularity (incremental programming, proper modifiers [public / private]): 10%
   - Proper Indentation: 5%
   - Meaningful Identifiers (for both method and variable names): 5%

   Compilation Error **:** Deduction of **50% of the total marks obtained**.