8. Smoothing an Image

Program Name: Smooth.java Input File: smooth.dat

An image can be represented by a rectangular grid of pixels. Each pixel has associated with it a triplet of numbers giving the intensities of the red, green, and blue colors. These intensities are on a scale of 0 to 255. There are times that an image needs to be smoothed because of local imperfections in the image. The smoothing is done by replacing the offending pixel with the average value of the pixels surrounding that pixel.

This is a different version of the smoothing problem. The image will be a square matrix of numbers in the range 0 to 255. One intensity value will be associated with each pixel instead of three. The whole image will be smoothed instead of just particular regions in the image. The sub-grid over which the averaging is to be done will be specified in the problem. The sub-grid will be a square of odd dimension so that the pixel that is to be replaced will be at the center of the sub-grid. Obviously, the sub-grid will get truncated at the edges of the image. The average will include the value of the central pixel and is rounded to the closest integer value.

Input

The first line of input will contain two integers `n` and `m`:

- The integer `n` indicates the dimension of the square image.
- The integer `m` is odd and smaller than `n` and indicates the dimension of the sub-grid over which the average has to be taken .
- Each of the next `n` lines of data will contain `n` integer numbers in the range 0 to 255 separated by one or more spaces.

Output

You will print out the smoothed image. The smoothed image will be `n` lines of data each line having `n` integer values in the range 0 to 255 followed by a single space.

Example Input File

```
10 3
  65  223  255  133  221   95  141   41  172  127
 177   37   68    0  224  196  243  145   61   75
 236  151  207  197   41  106  120  216  215  159
 226   57  176   30  224   67  217  244  246   22
 226   57   27   31   46  101  250  255  234  160
 100  140  250  184   73  206   90  212  131    9
 109  147  116  226  217  238  117  244  187  198
  24   19   86  162    5  227  189    1   41   21
  30   49  169  238  149  158  112   87  206  211
 181  112   54  199  196  106  174   63    6   73
```

Example Output to Screen

```
126 138 119 150 145 187 144 134 104 109
148 158 141 150 135 154 145 150 135 135
147 148 103 130 121 160 173 190 154 130
159 151 104 109 94 130 175 222 195 173
134 140 106 116 107 142 182 209 168 134
130 130 131 130 147 149 190 191 181 153
90 110 148 147 171 151 169 135 116 98
63 83 135 152 180 157 153 132 133 144
69 80 121 140 160 146 124 98 79 93
93 99 137 168 174 149 117 108 108 124
```