Introduction to Computer Vision

Jojo is an outstanding student who is currently pursuing his CS degree in Binus University. In one of his Computer Vision lectures, his lecturer tackled a topic about binary thresholding.

For simplicity, binary thresholding is an image filtering method used to separate low valued pixels (dark) from high valued pixels (bright). Each pixels in an image is classified as low or high based on a certain threshold K. If a pixel $P_{ij} \leq K$, then it is classified as low pixel, otherwise it is considered as high pixel.

The problem with binary thresholding method is that the value of K may varies from images. If K is set too a very low value, then a lot of pixels in the image will be considered high valued pixels, likewise the opposite case. Thus Jojo wants to find the value of K such that the absolute difference between the amount of dark pixels and bright pixels is minimum. Jojo thinks that the most optimal value of K is the data's median. Help Jojo to find the value K he wants.

Format Input

The program is expected to read a file "testdata.in" which contains at most 500 test cases of inputs, each containing 2 integers N, M which defines dimension of the image. Each test case contains N lines of M integers, where each value represents the image pixel value.

Format Output

Output should be expressed in format "Case #X: Y" - X is the number of the test case, and followed by Y rounded to the nearest integer number, such that when binary threshold Y is applied to the image, the absolute difference between the amount of dark pixels and bright pixels in the filtered image is minimum.

Constraints

- $1 \leq T \leq 500$
- $1 \leq N, M \leq 100$
- $0 \leq P_{ij} \leq 255$

Sample Input (testdata.in)

```
2
19 3
1 235 160
2 159 113
162 46 106
108 216 124
22 189 16
60 119 185
156 16 239
76 123 113
83 60 25
52 3 80
184 233 53
168 162 135
6 9 214
124 250 200
195 49 48
8 13 164
91 236 108
229 166 176
58 107 43
7 11
169 127 197 173 160 172 132 202 25 5 79
111 17 106 100 120 231 32 218 43 218 88
238 80 172 106 94 204 102 143 185 160 19
32 1 215 244 178 109 246 110 54 193 89
163 212 229 95 52 7 54 146 61 105 232
44 14 229 54 2 241 22 21 43 44 39
34 235 97 138 200 195 227 106 16 79 167
```

Sample Output (standard output)

```
Case #1: 113
Case #2: 109
```

Introduction to Computer Vision

Jojo merupakan murid yang cerdas dan sekarang sedang menempuh perkuliahan Sarjana Teknik Informatika di Binus University. Dalam salah satu mata kuliah, *Computer Vision*, sang dosen menjelaskan materi tentang *binary thresholding*.

Singkatnya, *binary thresholding* merupakan sebuah filter gambar sederhana untuk memisahkan *low valued pixels* atau warna gelap dari warna terang atau *high valued pixels*. Setiap pixel dari sebuah gambar bisa digolongkan sebagai warna gelap ataupun terang berdasarkan suatu nilai K tertentu. Jika suatu nilai pixel $P_{ij} \leq K$, maka bisa dikategorikan sebagai pixel gelap, dan selain itu bisa dikategorikan sebagai pixel terang.

Permasalahan dalam metode *binary thresholding* adalah nilai *threshold* K untuk setiap gambar bisa berbeda-beda. Jika nilai K diatur terlalu rendah, maka akan terdapat banyak pixel dalam gambar yang dianggap sebagai *high valued pixels*, dan juga untuk kasus sebaliknya. Oleh karenanya, Jojo ingin mencari sebuah nilai K untuk setiap gambar yang ia miliki sehingga perbedaan absolut antara pixel gelap dan terang menjadi seminimum mungkin. Jojo berpikir bahwa nilai K paling optimal adalah median dari data. Bantulah Jojo dalam mencari nilai K yang diinginkan.

Format Input

Program diharapkan dapat membaca sebuah file "testdata.in" yang berisi paling banyak 500 kasus uji, yang setiap kasusnya terdapat 2 angka bulat N dan M sebagai dimensi dari gambar. Setiap kasus uji mengandung N baris yang berisi M buah angka bulat, yang mana setiap nilai merepresentasikan nilai pixel pada gambar.

Format Output

Output yang dikeluarkan dalam format "Case #X: Y" - X merupakan nomor *test case*, dan diikuti oleh Y dibulatkan ke bilangan terdekat, sehingga untuk nilai *binary threshold* Y apabila diimplementasikan pada gambar ke X, nilai perbedaan absolut antara pixel gelap dan terang akan menjadi minimum.

Constraints

- $1 \leq T \leq 500$
- $1 \leq N, M \leq 100$
- $0 \leq P_{ij} \leq 255$

Sample Input (testdata.in)

```
2
19 3
1 235 160
2 159 113
162 46 106
108 216 124
22 189 16
60 119 185
156 16 239
76 123 113
83 60 25
52 3 80
184 233 53
168 162 135
6 9 214
124 250 200
195 49 48
8 13 164
91 236 108
229 166 176
58 107 43
7 11
169 127 197 173 160 172 132 202 25 5 79
111 17 106 100 120 231 32 218 43 218 88
238 80 172 106 94 204 102 143 185 160 19
32 1 215 244 178 109 246 110 54 193 89
163 212 229 95 52 7 54 146 61 105 232
44 14 229 54 2 241 22 21 43 44 39
34 235 97 138 200 195 227 106 16 79 167
```

Sample Output (standard output)

```
Case #1: 113
Case #2: 109
```