Count submatrices 3x3 with all digits 1, 2, ...,9 X63887_en

Given a matrix with digits 1, 2, ..., 9, you must count how many submatrices of size 3×3 of the given matrix contain all different digits 1, 2, ..., 9.

Input

The input has several cases. Each one starts with two positive natural numbers n, m in the first line. Afterwards comes a matrix of size $n \times m$ with digits 1, 2, ..., 9 (n lines with m digits each).

Output

For each case, the program must write in a new line the number of submatrices 3×3 inside of which all digits 1, 2, ..., 9 appear.

Sample input 1

```
3 3
284
379
516
2 3
568
657
7 5
19847
36269
74513
96812
14736
32594
96812
2 6
111137
367687
1 3
647
12 3
389
571
624
754
361
829
457
893
621
475
639
281
7 6
728738
631924
954651
619853
473291
689674
251434
4 1
1
3
2
8
8 6
563647
489823
127519
463476
895614
938927
574583
216272
9 12
923936738927
714541521658
568728946341
932837671279
816717885981
475942586438
748196371741
326324196856
159758252932
8 12
485369419783
162123743645
379451885219
887694628435
521924259768
734581948459
986736367132
729122215786
```

```
7 3
963
725
184
412
675
938
142
5 3
289
573
146
893
527
4 4
1275
4834
5691
1275
3 10
5825239638
1674148275
4931675149
2 1
3
3
11 3
685
123
749
658
123
984
657
845
217
936
458
12 7
9346874
8262392
1578561
3691478
8243751
7587932
9429486
3613517
7587627
1429539
5374184
8963857
1 3
748
5 7
5246273
6383518
1971964
5625281
4384735
1 11
67748253494
```

Sample output 1

```
1
0
8
0
0
6
8
0
9
24
19
3
2
4
6
0
6
21
0
7
0
```

Sample input 2

```
4 19
2686421479639762971
3751576365295341791
1498398128341283583
8623241479678686426
5 10
6253547831
3189389576
7495621942
6251475831
1383983756
15 10
5146514219
5892857819
1971396326
4854639457
2632157182
8578975963
3213468475
9649231286
8125471753
6741965941
3952832813
7683617637
8954954184
6712739259
2348167815
15 11
53297347172
46151851358
98742692469
19517437271
87435816468
36283775953
85769138261
91424574945
67889381387
32571626563
79577543891
68326136472
24162848598
89757951236
35614359174
3 9
462374976
315465814
978912325
13 10
5176892848
3621541578
4893473693
3829298241
1973749319
4562696752
7841852486
6397543137
9432628592
8619719361
2758467473
4393182586
8162793192
15 14
46784329654869
35139574729172
82941861832534
38676597249239
79848234756786
41271549813541
56359762492913
92648813712415
84723411465392
51382978398768
26958567256145
64272392174682
92435169496714
87697487832539
35182325715862
16 20
87978515975325386359
12463972484951975614
56312463821676421278
79851851963823589395
47243457547216352416
81651825363591976188
39567964764864283947
49332489139359563136
76515832581277319258
21821295349315943862
34925646276862865715
92787389815293592439
58614934517547183155
41353483849816428698
66926925623265935712
44781757915478712435
19 10
8531593462
2648728251
9713641976
3859218348
9426659617
6173347259
5323858241
9482699538
5241741679
3793587369
1685192547
5376436821
4693267382
8218853195
3754912476
3729467613
9548944937
8613255826
4297163154
4 11
28192795719
95433656345
37618419826
99282795719
5 6
```

247585
631258
589631
728479
855127
19 4
1541
9327
7685
5941
6723
4317
7694
5285
8167
7349
2596
8612
1688
9429
7356
8718
6942
3525
8711
20 5
26492
39863
71571
93243
75695
41871
62365
75947
41218
38637
79549
26397
78168
94524
97937
83561
94284
61793
15156
49284
18 6
137266
689349
425186
174527
613613
728975
945675
678934
213218
428765
396169
571425
762837
835961

941743
276958
538327
491614
9 14
89392326587687
98148159462419
43756784319352
56293581587547
89287437426874
45795692137532
31632186529619
29418793684827
87597524317354
14 5
14519
73874
59263
64159
78371
37637
28428
95195
47386
86247
45875
72943
13612
45875
10 9
578441371
419526732
632698546
981241923
372893871
465675461
832839829
719574573
276421633
421389379
7 5
54328
98197
27654
92344
64531
82762
31984
8 13
5672617578152
9839182424683
2145863935947
5672696414364
2839175357951
1493218262815
2751379714369
3682564531427
20 6
978612
132378
654594

```
271375
496286
385419
546976
918231
237548
456197
462825
891463
573596
965491
342852
871376
596491
378587
412362
596491
```

Sample output 2

```
14
11
31
31
5
27
38
60
37
8
5
13
22
22
27
19
15
7
22
27
```

Observation

There is no need to optimize this problem. Any reasonably efficient implementation will pass.
Evaluation over 10 points:

- Passes public test cases: 5 points.
- Passes public and private test cases: 10 points.

Problem information

Author: PRO1

Generation: 2026-01-25T21:58:11.621Z