CSCD 439/539 GPU Computing Lab3
Map 2D Thread Grid to Data

No Late Submissions are accepted. **Rules:** Your code must use C and CUDA Language. If your program shows a compilation error, you get a zero for this lab assignment.

Submission: Wrap up all your **source files** into a single zip file. Name your zip file as *FirstInitialYourLastName*Lab3.zip. For example, if your legal name is Will Smith, you should name your zip file as wsmithlab3.zip. A simple makefile has been provided in the zip file.

Before you leave the laboratory, please show the TA or the instructor how your program works, they will give you a score for this Lab assignment.

For archive purpose, please also submit your single zip file on EWU Canvas by following CSCD439-01 Course → Assignments → Lab3 → Submit Assignment to upload your single zip file.

Problem Description:

Based on the last lecture about how to map 2D grid to 2D dataset, and about store 2D matrix as a flattened 1D array, you are required to implement the following features and answer the questions.

In the program, we have a 16-by-16 2D matrix **A** as input. It is harder to setup 2D arrays or double pointers on GPU device (we learn how to do pointers to pointers on GPU this week!). In this lab, **A** is linearized and stored in 1D array on device by using row-major storage. I have set up execution configuration variables where the block size in **x** direction is 3 and block size in the **y** direction is 4.

1, Read the provided code and understand the input, the output, and how data is transferred between host and device.

2, According to the execution configurations, please write a kernel named **kernel2** to produce the following output. In kernel, you are required to ONLY use a subset of the build-in variables, blockIdx.x, blockDim.x, threadIdx.x, threadIdx.y, gridDim.x or gridDim.y, or some intermediate results from the combinations of these build-in variables. E.g. **int ix = blockIdx.x * blockDim.x + threadIdx.x** is allowed to use. Formal parameters of the kernel function are also allowed to use.

```
0   0   0   1   1   1   2   2   2   3   3   3   4   4   4   5
0   0   0   1   1   1   2   2   2   3   3   3   4   4   4   5
0   0   0   1   1   1   2   2   2   3   3   3   4   4   4   5
0   0   0   1   1   1   2   2   2   3   3   3   4   4   4   5
6   6   6   7   7   7   8   8   8   9   9   9   10  10  10  11
6   6   6   7   7   7   8   8   8   9   9   9   10  10  10  11
6   6   6   7   7   7   8   8   8   9   9   9   10  10  10  11
6   6   6   7   7   7   8   8   8   9   9   9   10  10  10  11
12  12  12  13  13  13  14  14  14  15  15  15  16  16  16  17
12  12  12  13  13  13  14  14  14  15  15  15  16  16  16  17
12  12  12  13  13  13  14  14  14  15  15  15  16  16  16  17
12  12  12  13  13  13  14  14  14  15  15  15  16  16  16  17
18  18  18  19  19  19  20  20  20  21  21  21  22  22  22  23
18  18  18  19  19  19  20  20  20  21  21  21  22  22  22  23
18  18  18  19  19  19  20  20  20  21  21  21  22  22  22  23
18  18  18  19  19  19  20  20  20  21  21  21  22  22  22  23
```

3, According to the execution configurations, please write a kernel named **kernel3** to produce the following output. In kernel, you are required to ONLY use a subset of the build-in variables, blockIdx.x, blockDim.x, threadIdx.x, threadIdx.y, gridDim.x or gridDim.y, or some intermediate results from the combinations of these build-in variables. E.g. **int ix = blockIdx.x * blockDim.x + threadIdx.x** is allowed to use. Formal parameters of the kernel function are also allowed to use.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

4, According to the execution configurations, please write a kernel named **kernel4** to produce the following output. In kernel, you are required to ONLY use a subset of the build-in variables, blockIdx.x, blockDim.x, threadIdx.x, threadIdx.y, gridDim.x or gridDim.y, or some intermediate results from the combinations of these build-in variables. E.g. **int ix = blockIdx.x * blockDim.x + threadIdx.x** is allowed to use. Formal parameters of the kernel function are also allowed to use.

0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6	7	8	6	7	8	6
9	10	11	9	10	11	9	10	11	9	10	11	9	10	11	9
0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6	7	8	6	7	8	6
9	10	11	9	10	11	9	10	11	9	10	11	9	10	11	9
0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6	7	8	6	7	8	6
9	10	11	9	10	11	9	10	11	9	10	11	9	10	11	9
0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6	7	8	6	7	8	6
9	10	11	9	10	11	9	10	11	9	10	11	9	10	11	9

5, According to the execution configurations, please write a kernel named **kernel5** to produce the following output. In kernel, you are required to ONLY use a subset of the build-in variables, blockIdx.x, blockDim.x, threadIdx.x, threadIdx.y, gridDim.x or gridDim.y, or some intermediate results from the combinations of these build-in variables. E.g. **int ix = blockIdx.x * blockDim.x + threadIdx.x** is allowed to use. Formal parameters of the kernel function are also allowed to use.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

6, According to the execution configurations, please write a kernel named **kernel6** to produce the following output. In kernel, you are required to ONLY use a subset of the build-in variables, blockIdx.x, blockDim.x, threadIdx.x, threadIdx.y, gridDim.x or gridDim.y, or some intermediate results from the combinations of these build-in variables. E.g. **int ix = blockIdx.x * blockDim.x + threadIdx.x** is allowed to use. Formal parameters of the kernel function are also allowed to use.

0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5

7, Please look into the provided zip file, and implement all your kernel functions in the **myKernel.cu** source file. Please do not change other part of provided program. Instead, you are encouraged to understand all features in the provided zip file.