Name: ______________________

Lab for ITSC 3181 and 3181L, Introduction to Computer Architecture, Spring 2020

Lab #13 and #14: Memory and Cache Access, Stride Access and Loop Ordering with Matrix Multiplication

Grade:

	Task 1 and 2 (75%)					Task 3 (25%)			Total
	ikj (multMat2)	jik (multMat3)	jki (multMat4)	kij (multMat5)	kji (multMat6)	Question 1	Question 2	Question 3	
Percentage	15%	15%	15%	15%	15%	8%	8%	9%	
Your grade									

Task 1 and 2 (remove this figure and paste yours that has the solution)

2	ijk (multMat1)																					
3		k=0		k=1		k=2		...	k=7		k=8		...	k=62		k=63		Summary				
4	Array	Element	Cache H	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	Stride	# Acces	# Miss	Miss Ra	Notes
5	Load for A	A[0]	M	A[n]	M	A[2n]	M	...	A[7n]	M	A[8n]	M	...	A[62n]	M	A[63n]	M	n	64	64	100%	Every access is Miss
6	Load for B	B[0]	M	B[1]	H	B[2]	H	...	B[7]	H	B[8]	M	...	B[62]	H	B[63]	H	1	64	8	0.13	1 out of 8 accesses is miss. E
7	Load for C	C[0]	M	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	0	64	1	0.02	For all the 64 accesses, the f
8	Store for C	C[0]	H	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	0	64	0	0	All 64 accesses are hit
9																		Total	256	73	0.29	Miss Rate = # Misses / # Acce

12	ikj (multMat2)																					
13		k=0		k=1		k=2		...	k=7		k=8		...	k=62		k=63		Summary				
14	Array	Element	Cache H	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	Stride	# Acces	# Miss	Miss Ra	Notes
15	Load for A	A[0]	M	A[1]	M	A[2]	M	...	A[7]	M	A[8]	M	...	A[62]	M	A[63]	M	1	64	64	100%	
16	Load for B	B[0]	M	B[n]	M	B[2n]	M	...	B[7n]	M	B[8n]	M	...	B[62n]	H	B[63n]	H	n	64	16	0.25	
17	Load for C	C[0]	M	C[jn]	M	C[2jn]	M	...	C[7jn]	M	C[8jn]	M	...	C[62jn]	H	C[63jn]	H	jn	64	8	0.13	
18	Store for C	C[0]	H	C[jn]	H	C[2jn]	H	...	C[7jn]	H	C[8jn]	H	...	C[62jn]	H	C[63jn]	H	jn	64	0	0	All 64 accesses are hit
19																		Total	256	88	0.34	Miss Rate = # Misses / # Acce

22	jik (multMat3)																					
23		k=0		k=1		k=2		...	k=7		k=8		...	k=62		k=63		Summary				
24	Array	Element	Cache H	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	Stride	# Acces	# Miss	Miss Ra	Notes
25	Load for A	A[0]	M	A[n]	M	A[2n]	M	...	A[7n]	M	A[8n]	M	...	A[62n]	M	A[63n]	M	n	64	64	100%	All are misses
26	Load for B	B[0]	M	B[1]	H	B[2]	H	...	B[7]	H	B[8]	M	...	B[62]	H	B[63]	H	1	64	8	0.13	1/8 are a miss
27	Load for C	C[0]	M	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	0	64	1	0.02	Only first access is a miss
28	Store for C	C[0]	H	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	0	64	0	0	All 64 accesses are hit
29																		Total	256	73	0.29	Miss Rate = # Misses / # Acce

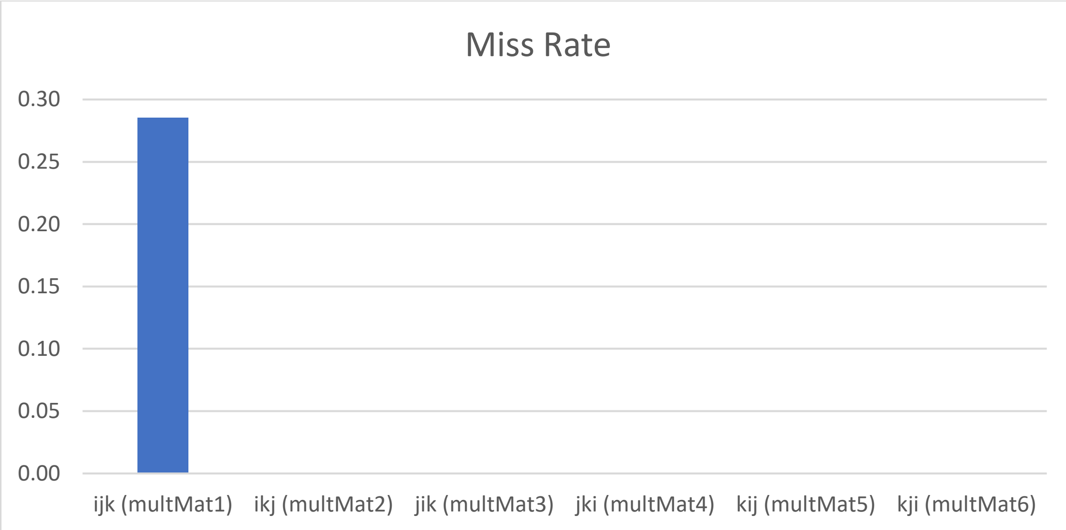
32	jki (multMat4)																					
33		k=0		k=1		k=2		...	k=7		k=8		...	k=62		k=63		Summary				
34	Array	Element	Cache H	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	Stride	# Acces	# Miss	Miss Ra	Notes
35	Load for A	A[0]	M	A[n]	M	A[2n]	M	...	A[7n]	M	A[8n]	M	...	A[62n]	H	A[63n]	H	n	64	8	13%	
36	Load for B	B[0]	M	B[1]	M	B[2]	M	...	B[7]	H	B[8]	H	...	B[62]	H	B[63]	H	1	64	2	0.03	
37	Load for C	C[0]	M	C[in]	M	C[2in]	H	...	C[7in]	H	C[8in]	H	...	C[62in]	H	C[63in]	H	in	64	1	0.02	
38	Store for C	C[0]	H	C[in]	H	C[2in]	H	...	C[7in]	H	C[8in]	H	...	C[62in]	H	C[63in]	H	in	64	0	0	All 64 accesses are hit
39																		Total	256	11	0.04	Miss Rate = # Misses / # Acce

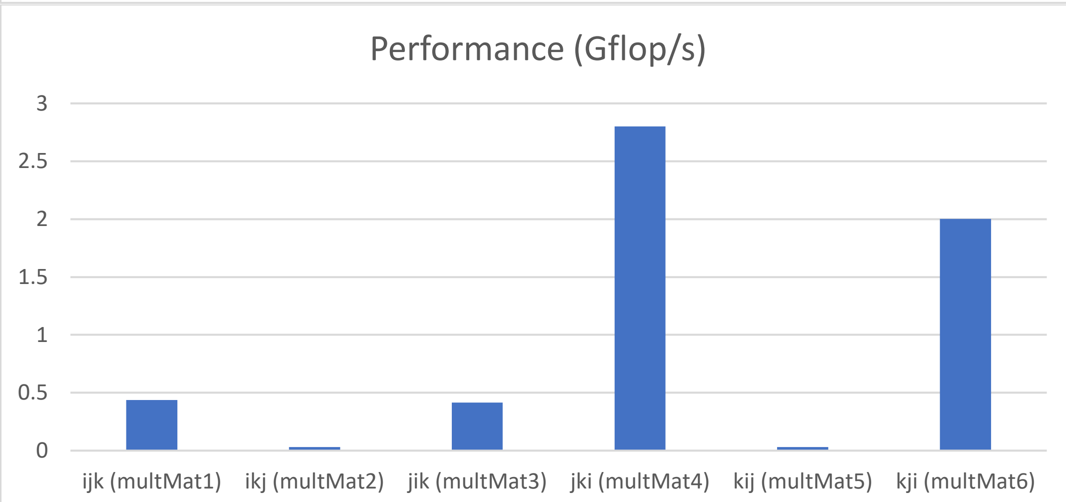
42	kij (multMat5)																					
43		k=0		k=1		k=2		...	k=7		k=8		...	k=62		k=63		Summary				
44	Array	Element	Cache H	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	Stride	# Acces	# Miss	Miss Ra	Notes
45	Load for A	A[0]	M	A[1]	M	A[2]	M	...	A[7]	M	A[8]	M	...	A[62]	M	A[63]	M	1	64	64	100%	
46	Load for B	B[0]	M	B[n]	H	B[2n]	H	...	B[7n]	H	B[8n]	H	...	B[62n]	H	B[63n]	H	n	64	0	0.00	
47	Load for C	C[0]	M	C[jn]	H	C[2jn]	H	...	C[7jn]	H	C[8jn]	H	...	C[62jn]	H	C[63jn]	H	jn	64	0	0.00	
48	Store for C	C[0]	H	C[jn]	H	C[2jn]	H	...	C[7jn]	H	C[8jn]	H	...	C[62jn]	H	C[63jn]	H	jn	64	0	0	All 64 accesses are hit
49																		Total	256	64	0.25	Miss Rate = # Misses / # Acce

52	kji (multMat6)																					
53		k=0		k=1		k=2		...	k=7		k=8		...	k=62		k=63		Summary				
54	Array	Element	Cache H	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	...	Element	Cache H	Element	Cache H	Stride	# Acces	# Miss	Miss Ra	Notes
55	Load for A	A[0]	M	A[n]	M	A[2n]	M	...	A[7n]	M	A[8n]	M	...	A[62n]	H	A[63n]	H	n	64	32	50%	
56	Load for B	B[0]	M	B[1]	M	B[2]	M	...	B[7]	M	B[8]	M	...	B[62]	H	B[63]	H	1	64	16	0.25	
57	Load for C	C[0]	M	C[0]	M	C[0]	H	...	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	0	64	1	0.02	
58	Store for C	C[0]	H	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	...	C[0]	H	C[0]	H	0	64	0	0	All 64 accesses are hit
59																		Total	256	49	0.19	Miss Rate = # Misses / # Acce

63		Miss Rate
64	ijk (multMat1	0.29
65	ikj (multMat2	0.34
66	jik (multMat3	0.29
67	jki (multMat4	0.04
68	kij (multMat5	0.25
69	kji (multMat6	0.19

Task 3	
	Performance (Gflop/s)
ijk (mult	0.437
ikj (mult	0.031
jik (mult	0.415
jki (mult	2.801
kij (mult	0.031
kji (mult	2.002





Task 2:

1. Which ordering(s) perform best for these 1000-by-1000 matrices? **Why?**

jki performs the best because it has the highest Gflop/s at 2.801 and kji is close in secons with 2.002 Gflop/s

2. Which ordering(s) perform the worst? **Why?**

ikj and kij both perform the worst because they have the lowest Gflop/s at 0.031

3. How does the way we stride through the matrices with respect to the innermost loop affect performance?

When the innermost loop is i it increments by 1 each time so it goes linearly. When it is j it is incremented by n which means that it jumps around a lot in the matrix. When it is k it does not change anything.