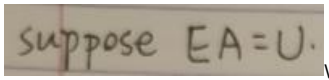
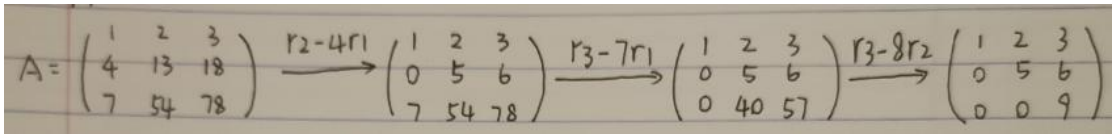
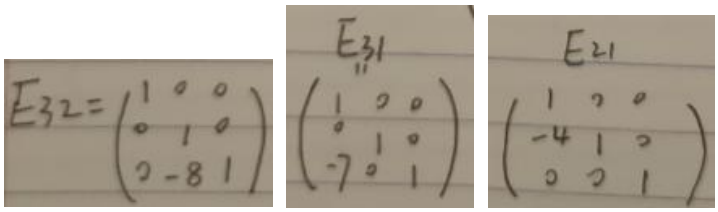
Hw2 Hairu Wen
GitHub Username: hwen020
https://github.com/CS211-Fall2023/hw2-hwen020
SID: 862467599

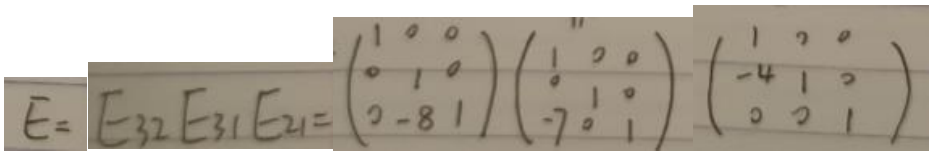
1.

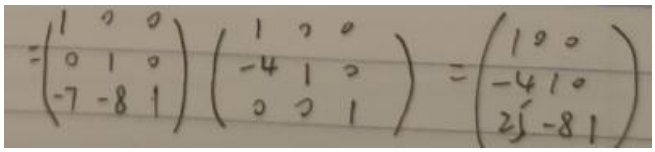


where E=E32*E31*E21.

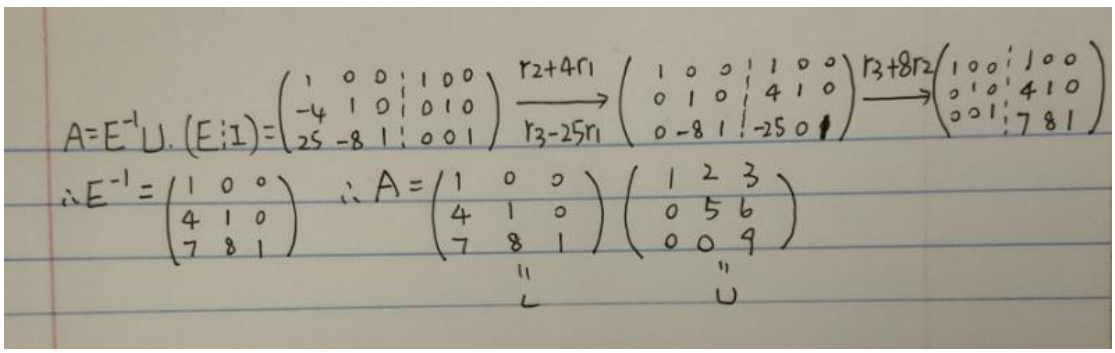








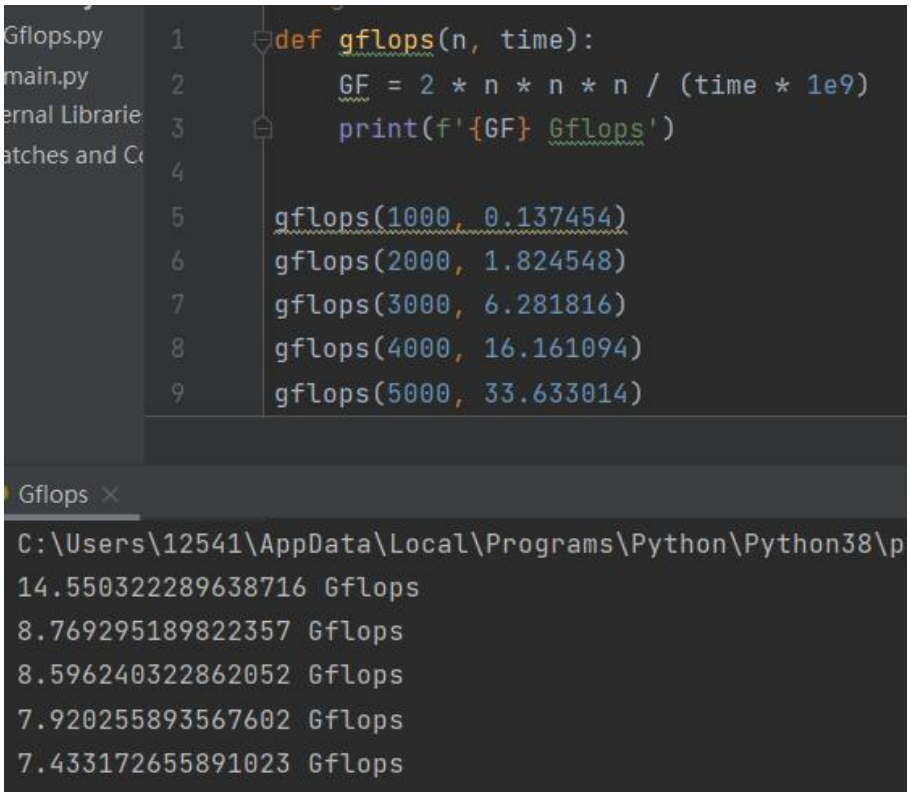
So



2. Test: implementation and LAPACK version with matrix size 1000, 2000, 3000, 4000, 5000. Compare the performance (i.e., Gflops) of the two approaches.

```
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 1000
n=1000, pad=1
time=0.137454s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 2000
n=2000, pad=1
time=1.824548s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 3000
n=3000, pad=1
time=6.281816s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 4000
n=4000, pad=1
time=16.161094s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 5000
n=5000, pad=1
time=33.633014s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 1000
n=1000, pad=1
time=0.078519s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 2000
n=2000, pad=1
time=0.266170s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 3000
n=3000, pad=1
time=0.775780s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 4000
n=4000, pad=1
time=1.503223s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 5000
n=5000, pad=1
time=3.041906s
```

[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 1000
n=1000, pad=1
time=0.137454s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 2000
n=2000, pad=1
time=1.824548s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 3000
n=3000, pad=1
time=6.281816s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 4000
n=4000, pad=1
time=16.161094s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my 5000
n=5000, pad=1
time=33.633014s



[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 1000
n=1000, pad=1
time=0.078519s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 2000
n=2000, pad=1
time=0.266170s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 3000
n=3000, pad=1
time=0.775780s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 4000
n=4000, pad=1
time=1.503223s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main lapack 5000
n=5000, pad=1
time=3.041906s

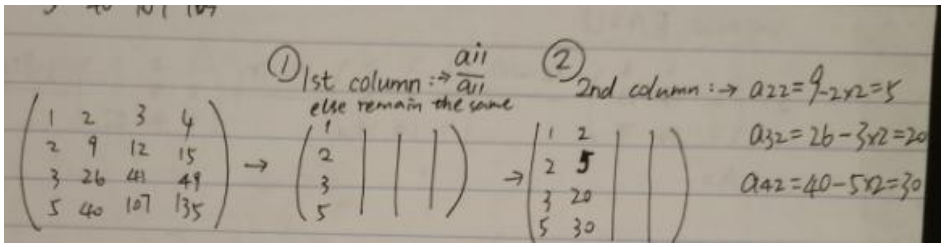
```
thonProject
Gflops.py
main.py
ternal Librarie
ratches and C
```

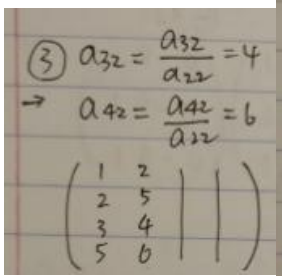
```python
def gflops(n, time):
    GF = 2 * n * n * n / (time * 1e9)
    print(f'{GF} Gflops')

gflops(1000, 0.078519)
gflops(2000, 0.266170)
gflops(3000, 0.775780)
gflops(4000, 1.503223)
gflops(5000, 3.041906)
```

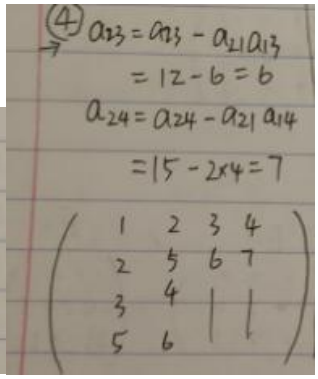
```
Gflops
C:\Users\12541\AppData\Local\Programs\Python\Python38\
25.471541919790113 Gflops
60.111958522748616 Gflops
69.60736291216583 Gflops
85.15037356400215 Gflops
82.18531407610887 Gflops
```

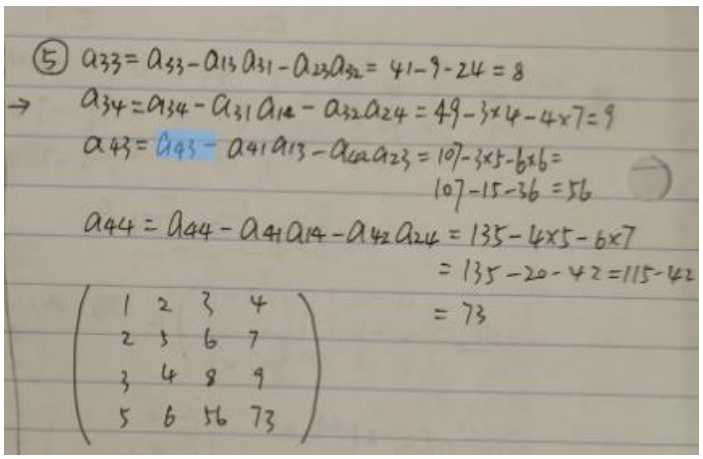
We can see from above that lapack has a better performance on this task.

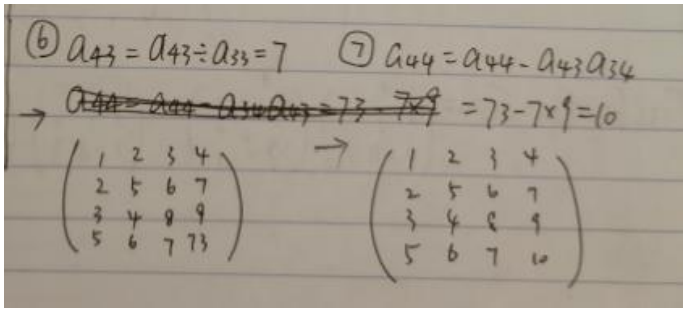
3.

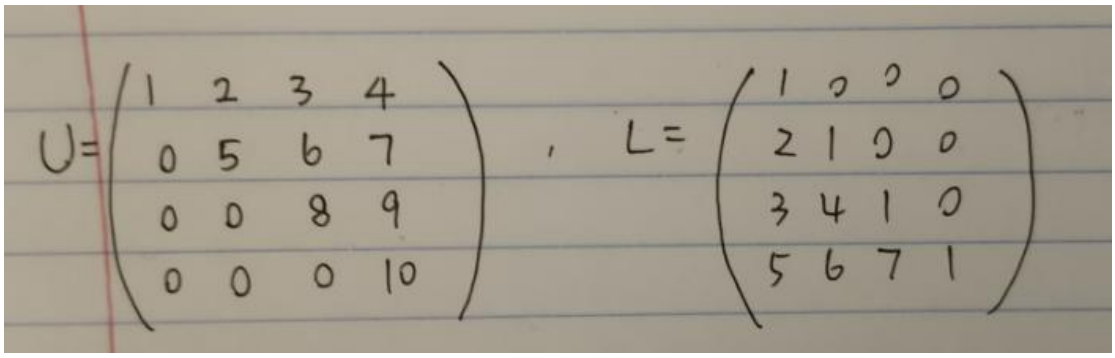












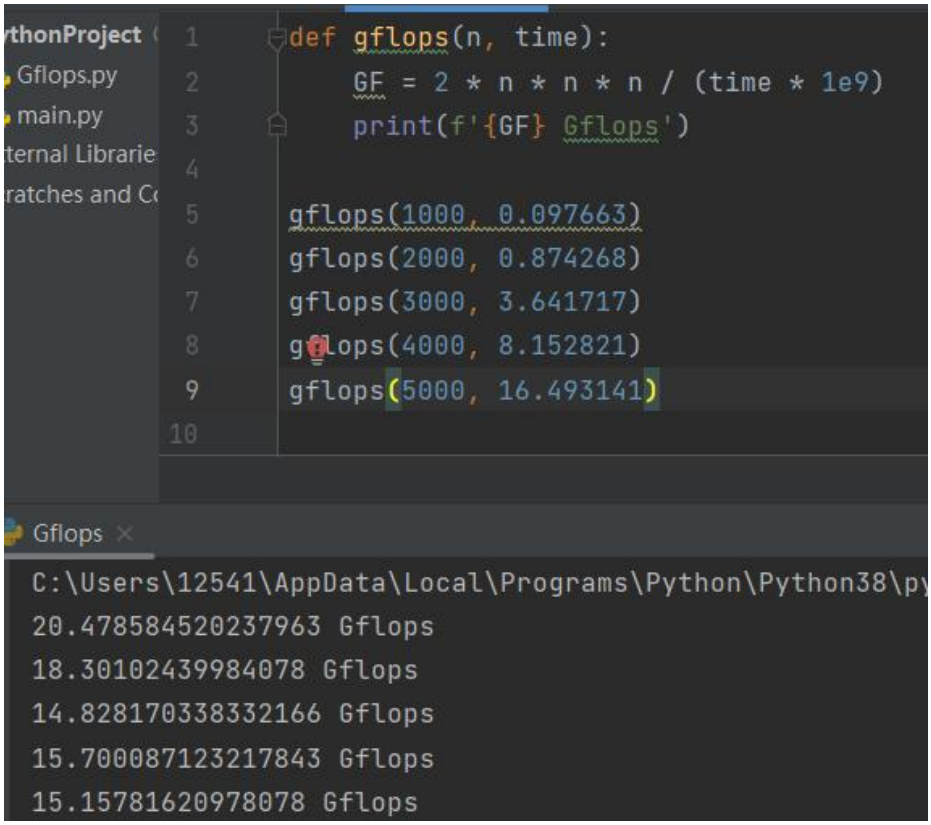
Bonus Point

When block_size=4:

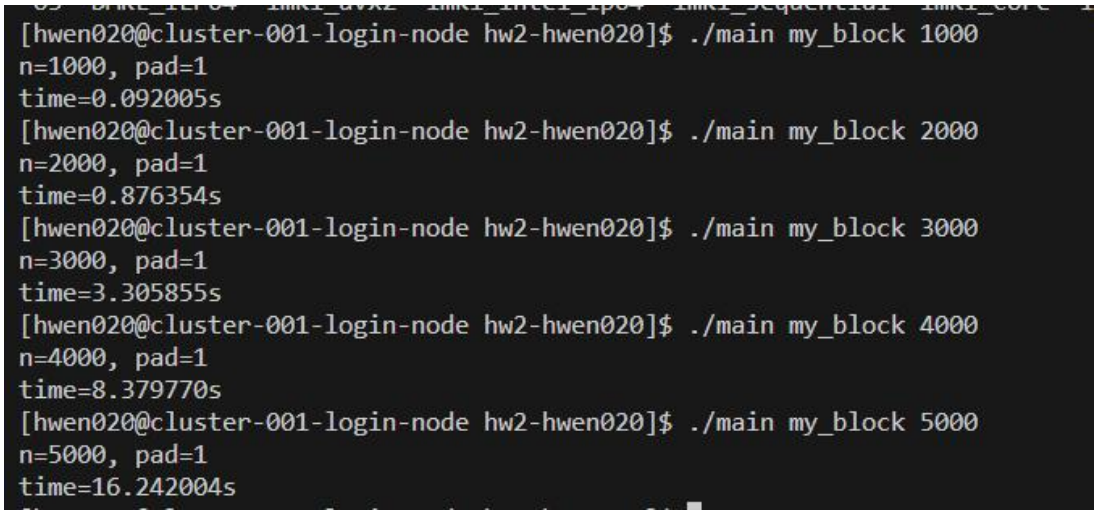
```
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 1000
n=1000, pad=1
time=0.097663s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 2000
n=2000, pad=1
time=0.874268s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 3000
n=3000, pad=1
time=3.641717s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 4000
n=4000, pad=1
time=8.152821s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 5000
n=5000, pad=1
time=16.493141s
```

[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 1000

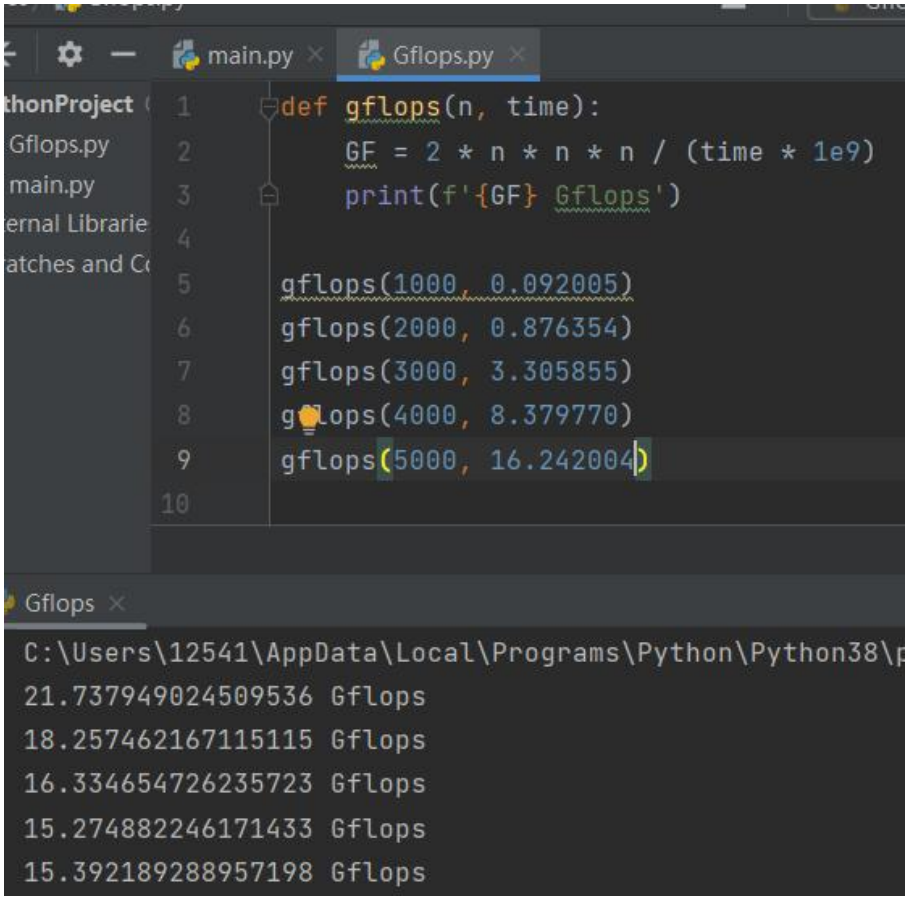
n=1000, pad=1
time=0.097663s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 2000
n=2000, pad=1
time=0.874268s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 3000
n=3000, pad=1
time=3.641717s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 4000
n=4000, pad=1
time=8.152821s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 5000
n=5000, pad=1
time=16.493141s



When block_size=2:



[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 1000
n=1000, pad=1
time=0.092005s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 2000
n=2000, pad=1
time=0.876354s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 3000
n=3000, pad=1
time=3.305855s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 4000
n=4000, pad=1
time=8.379770s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 5000
n=5000, pad=1
time=16.242004s



When block_size=20:

```
Already up to date.
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 1000
n=1000, pad=1
time=0.100313s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 2000
n=2000, pad=1
time=0.866703s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 3000
n=3000, pad=1
time=3.280529s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 4000
n=4000, pad=1
time=8.203220s
[hwen020@cluster-001-login-node hw2-hwen020]$ ./main my_block 5000
n=5000, pad=1
time=16.242142s
```

Seems block_size doesn't work for this.