



Lab Submission – 03

Arnab Mondal

20BCE1294

Program: B.Tech

Semester: Fall 2022-23

Course: CSE4001 – Parallel and Distributed Computing

Faculty: Dr. Sudha A

Date: 22-08-2022

Exercise: 03

Number of threads : 1

Matrix Size : 500

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 500

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i,j,k;
    double elapsed, start, end;
    omp_set_num_threads(1);
    for (i= 0; i< N; i++)
        for (j= 0; j< N; j++)
```

```

{
    A[i][j] = 2;
    B[i][j] = 2;
}

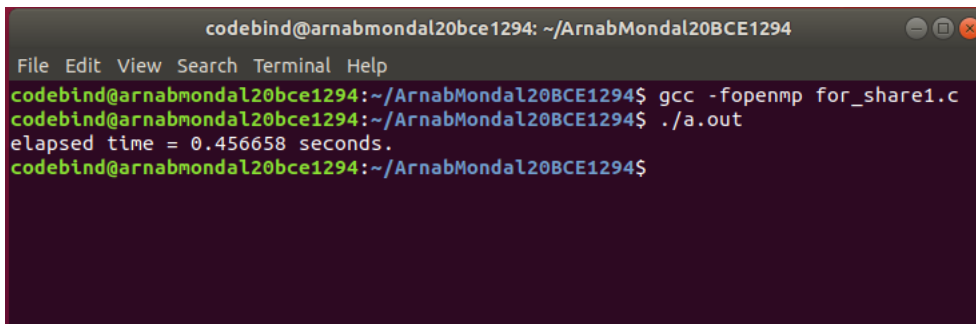
start = omp_get_wtime();
#pragma omp parallel for private(i,j,k) shared(A,B,C)
//schedule(static)
for (i = 0; i < N; ++i) {
    for (j = 0; j < N; ++j) {
        for (k = 0; k < N; ++k) {
            C[i][j] += A[i][k] * B[k][j];
        }
    }
}

end = omp_get_wtime();

elapsed = end - start;
printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:



```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 0.456658 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

Number of threads : 1

Matrix Size : 1000

Code:

```

#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

```

```

#include <sys/time.h>

#define N 1000

int A[N][N];

int B[N][N];

int C[N][N];

int main()
{
    int i, j, k;

    double elapsed, start, end;

    omp_set_num_threads(1);

    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;

            B[i][j] = 2;
        }

    start = omp_get_wtime();

#pragma omp parallel for private(i, j, k) shared(A, B, C)

    // schedule(static)

    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
}

```

```

    }
}

end = omp_get_wtime();

elapsed = end - start;

printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 4.596182 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

Number of threads : 4

Matrix Size : 1000

Code:

```

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 1000

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(4);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {

```

```

        A[i][j] = 2;
        B[i][j] = 2;
    }

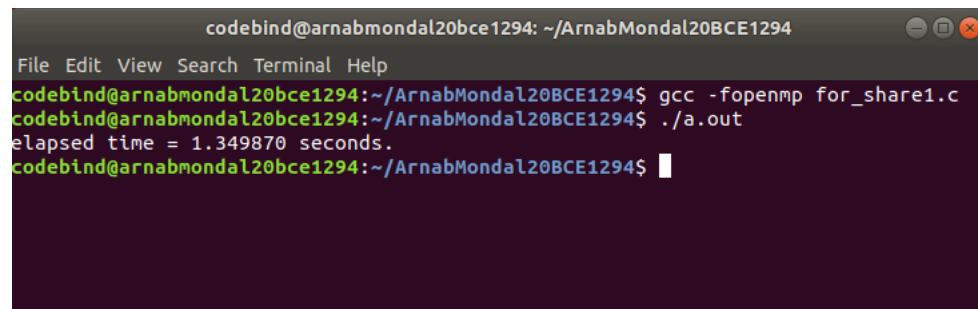
    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C)
    // schedule(static)
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:



```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 1.349870 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

Number of threads : 5

Matrix Size : 1000

Code:

```

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

```

```

#define N 1000

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(5);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;
        }

    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C)
    // schedule(static)
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 1.486515 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

Number of threads : 6

Matrix Size : 1000

Code:

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 1000

int A[N][N];
int B[N][N];
int C[N][N];

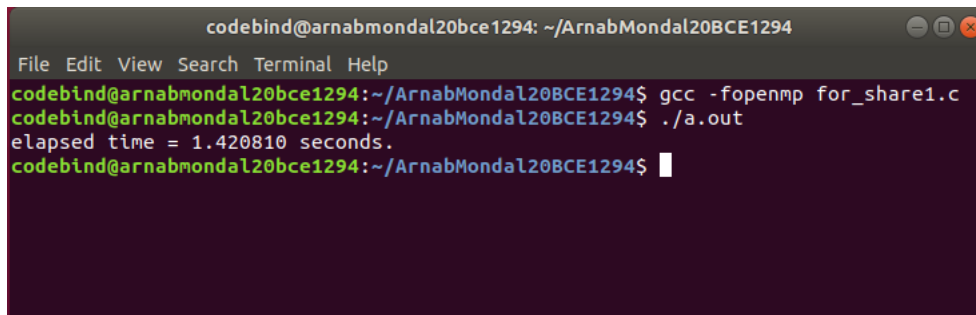
int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(6);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;
        }

    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C)
    // schedule(static)
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}
```

Output:



```
codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 1.420810 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$
```

Number of threads : 7

Matrix Size : 1000

Code:

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 1000

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(7);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;
        }

    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C)
    // schedule(static)
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
```



```

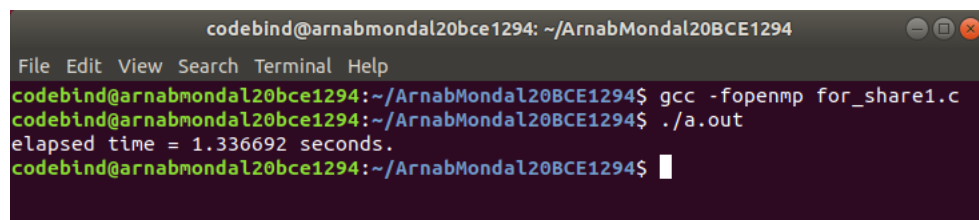
    {
        for (k = 0; k < N; ++k)
        {
            C[i][j] += A[i][k] * B[k][j];
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:



```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 1.336692 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

Number of threads : 8

Matrix Size : 1000

Code:

```

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 1000

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(8);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)

```

```

    {
        A[i][j] = 2;
        B[i][j] = 2;
    }

    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C)
    // schedule(static)
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 1.390833 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

$$\text{Speedup for 4 threads} = \frac{4.596182}{1.349870} = 3.404907$$

$$\text{Speedup for 5 threads} = \frac{4.596182}{1.486515} = 3.091917$$

$$\text{Speedup for 6 threads} = \frac{4.596182}{1.420810} = 3.234902$$

$$\text{Speedup for 7 threads} = \frac{4.596182}{1.336692} = 3.438474$$

$$\text{Speedup for 8 threads} = \frac{4.596182}{1.390833} = 3.304625$$

Efficiency for 4 threads – $\frac{3.404907}{4} = 0.851226 = 85.12\%$

Efficiency for 5 threads – $\frac{3.091917}{5} = 0.618383 = 61.83\%$

Efficiency for 6 threads – $\frac{3.234902}{6} = 0.539150 = 53.91\%$

Efficiency for 7 threads – $\frac{3.438474}{7} = 0.491210 = 49.12\%$

Efficiency for 8 threads – $\frac{3.304625}{8} = 0.413078 = 41.30\%$

(use loop sharing, matrix multiplication,1000,8,static)

Code:

```
#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

#include <sys/time.h>

#define N 1000

int A[N][N];

int B[N][N];

int C[N][N];

int main()

{

    int i, j, k;

    double elapsed, start, end;

    omp_set_num_threads(8);

    for (i = 0; i < N; i++)

        for (j = 0; j < N; j++)

        {
```

```

        A[i][j] = 2;

        B[i][j] = 2;
    }

    start = omp_get_wtime();

#pragma omp parallel for private(i, j, k) shared(A, B, C) schedule(static)

    for (i = 0; i < N; ++i)
    {

        for (j = 0; j < N; ++j)
        {

            for (k = 0; k < N; ++k)
            {

                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

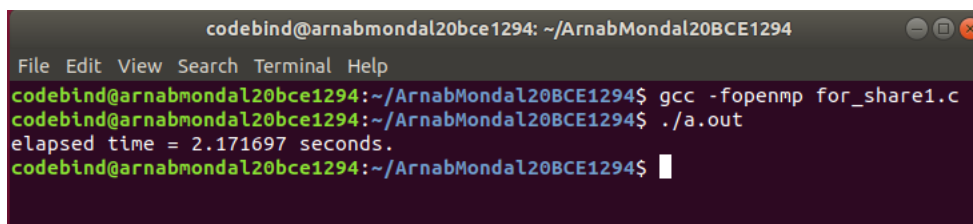
    end = omp_get_wtime();

    elapsed = end - start;

    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:



```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 2.171697 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

(use loop sharing, matrix multiplication,500,8,dynamic)

Code:

```

#include <pthread.h>

#include <stdio.h>

```

```
#include <stdlib.h>

#include <omp.h>

#include <sys/time.h>

#define N 500

int A[N][N];

int B[N][N];

int C[N][N];

int main()
{
    int i, j, k;

    double elapsed, start, end;

    omp_set_num_threads(8);

    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;

            B[i][j] = 2;
        }

    start = omp_get_wtime();

#pragma omp parallel for private(i, j, k) shared(A, B, C) schedule(dynamic)

    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
```

```

        C[i][j] += A[i][k] * B[k][j];
    }
}

end = omp_get_wtime();

elapsed = end - start;

printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share2.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 0.262950 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

(use loop sharing, matrix multiplication,500,8,dynamic,ordered)

Code:

```

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 500

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(8);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;

```

```

    }

    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C) schedule(dynamic)
ordered
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share2.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 0.262950 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gedit for_share2.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gedit for_share2.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share2.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 0.247714 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

(use loop sharing, matrix multiplication,1000,8,dynamic,ordered)

Code:

```

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 1000

int A[N][N];
int B[N][N];

```

```

int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(8);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;
        }

    start = omp_get_wtime();
#pragma omp parallel for private(i, j, k) shared(A, B, C) schedule(static)
    ordered
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```

codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 1.943430 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$

```

Using nowait and static

Code:

```

#include <pthread.h>
#include <stdio.h>

```



```

#include <stdlib.h>
#include <omp.h>
#include <math.h>
#include <sys/time.h>

#define N 1000

int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(8);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;
        }

    start = omp_get_wtime();
    // #pragma omp parallel for private(i,j,k) shared(A,B,C) schedule(static)
    // ordered nowait
    #pragma omp for private(i, j, k) schedule(static) ordered nowait
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    end = omp_get_wtime();

    elapsed = end - start;
    printf("elapsed time = %f seconds.\n", elapsed);
}

```

Output:

```
codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294
File Edit View Search Terminal Help
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share1.c
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out
elapsed time = 4.431207 seconds.
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$
```

Using nowait and dynamic

Code:

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>

#define N 500

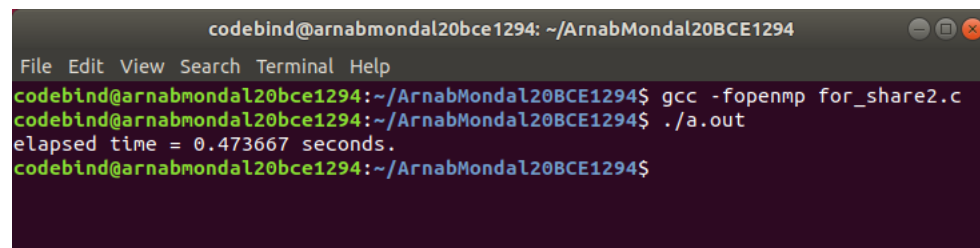
int A[N][N];
int B[N][N];
int C[N][N];

int main()
{
    int i, j, k;
    double elapsed, start, end;
    omp_set_num_threads(8);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
        {
            A[i][j] = 2;
            B[i][j] = 2;
        }

    start = omp_get_wtime();
    // #pragma omp parallel for private(i,j,k) shared(A,B,C) schedule(dynamic)
    // ordered nowait
    #pragma omp for private(i, j, k) schedule(dynamic) ordered nowait
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        {
            for (k = 0; k < N; ++k)
            {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
}
```

```
    }  
  }  
}  
  
end = omp_get_wtime();  
  
elapsed = end - start;  
printf("elapsed time = %f seconds.\n", elapsed);  
}
```

Output:



```
codebind@arnabmondal20bce1294: ~/ArnabMondal20BCE1294  
File Edit View Search Terminal Help  
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ gcc -fopenmp for_share2.c  
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$ ./a.out  
elapsed time = 0.473667 seconds.  
codebind@arnabmondal20bce1294:~/ArnabMondal20BCE1294$
```