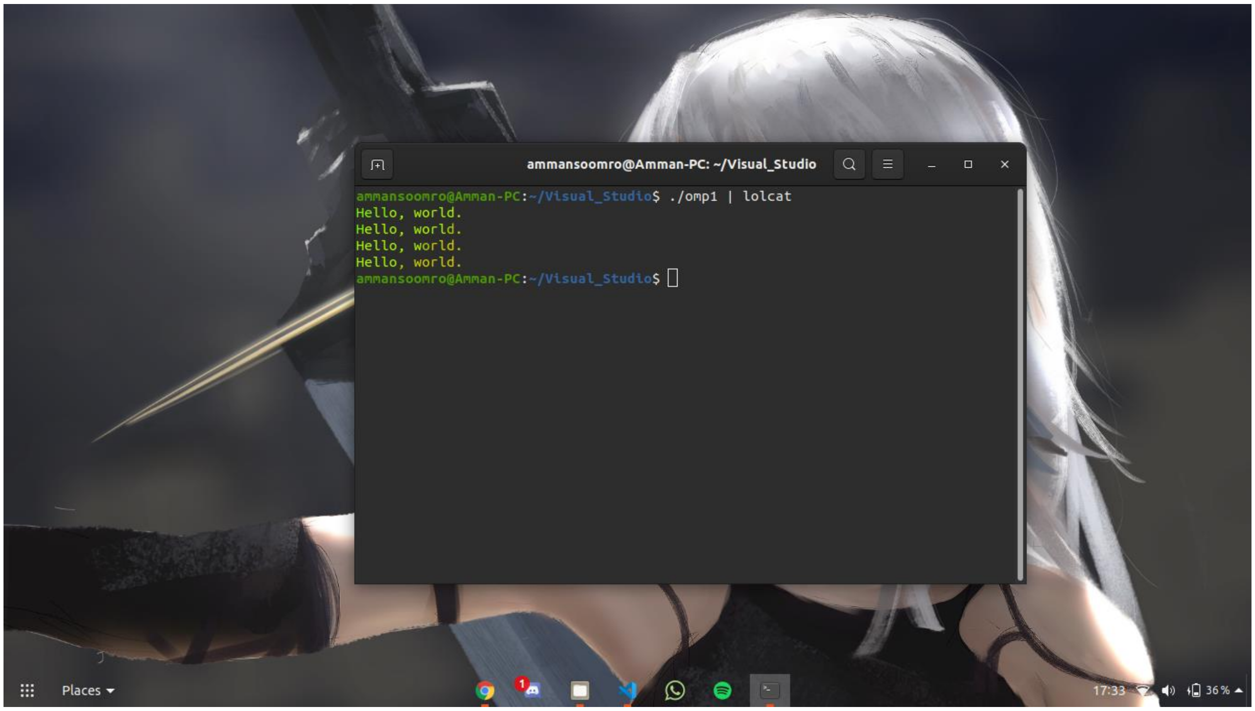
OS LAB 09 - OPENMP (K191048)

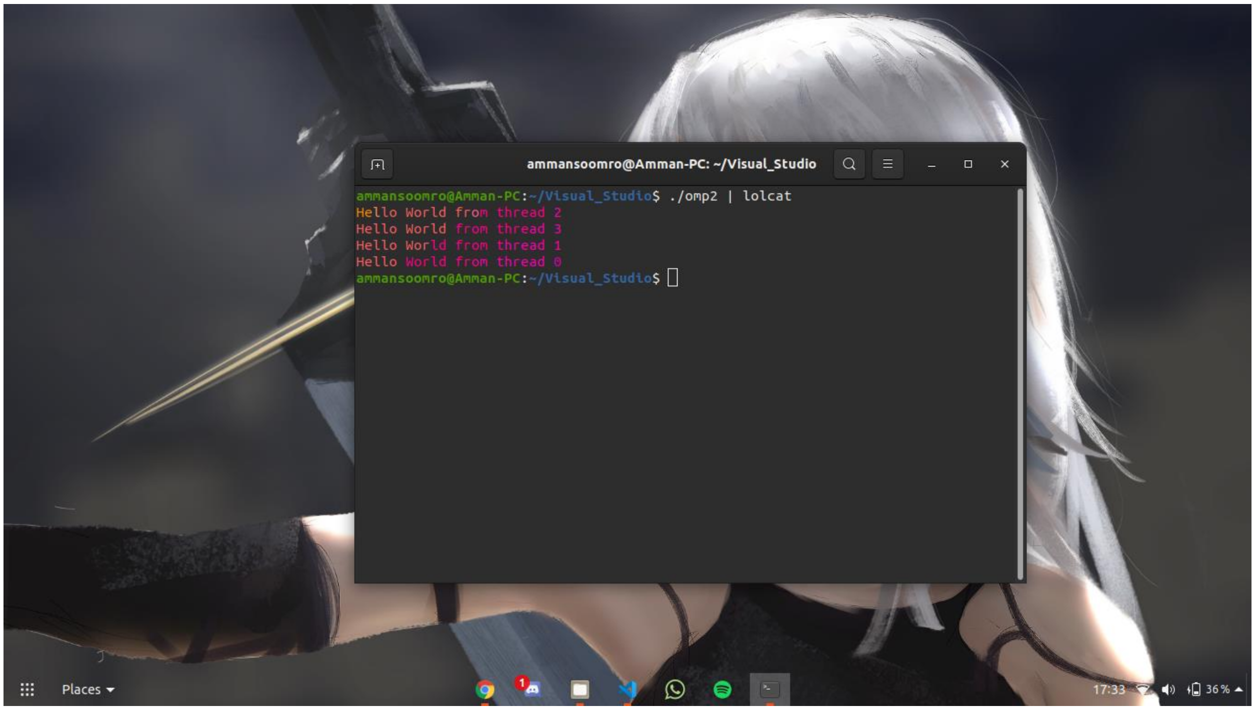
No code for the following screenshots since they are copied from lab activity.

OMP1



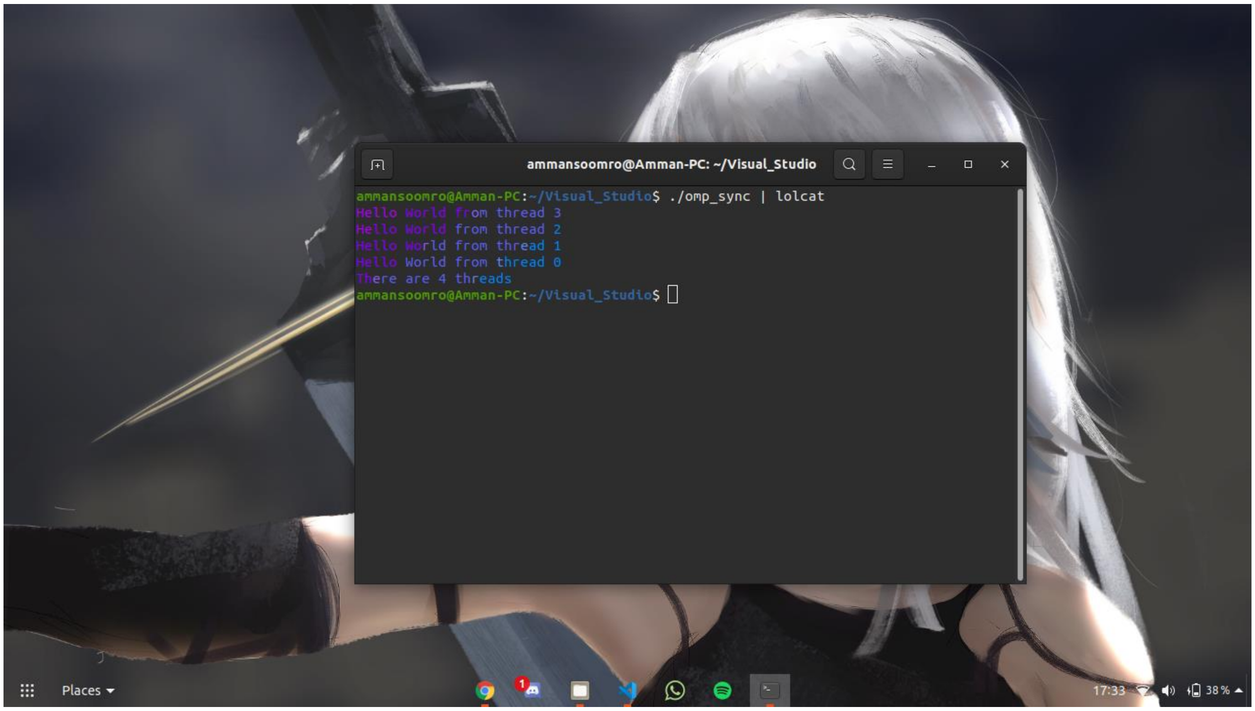
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp1 | lolcat
Hello, world.
Hello, world.
Hello, world.
Hello, world.
ammansoomro@Amman-PC:~/Visual_Studio$
```

OMP2



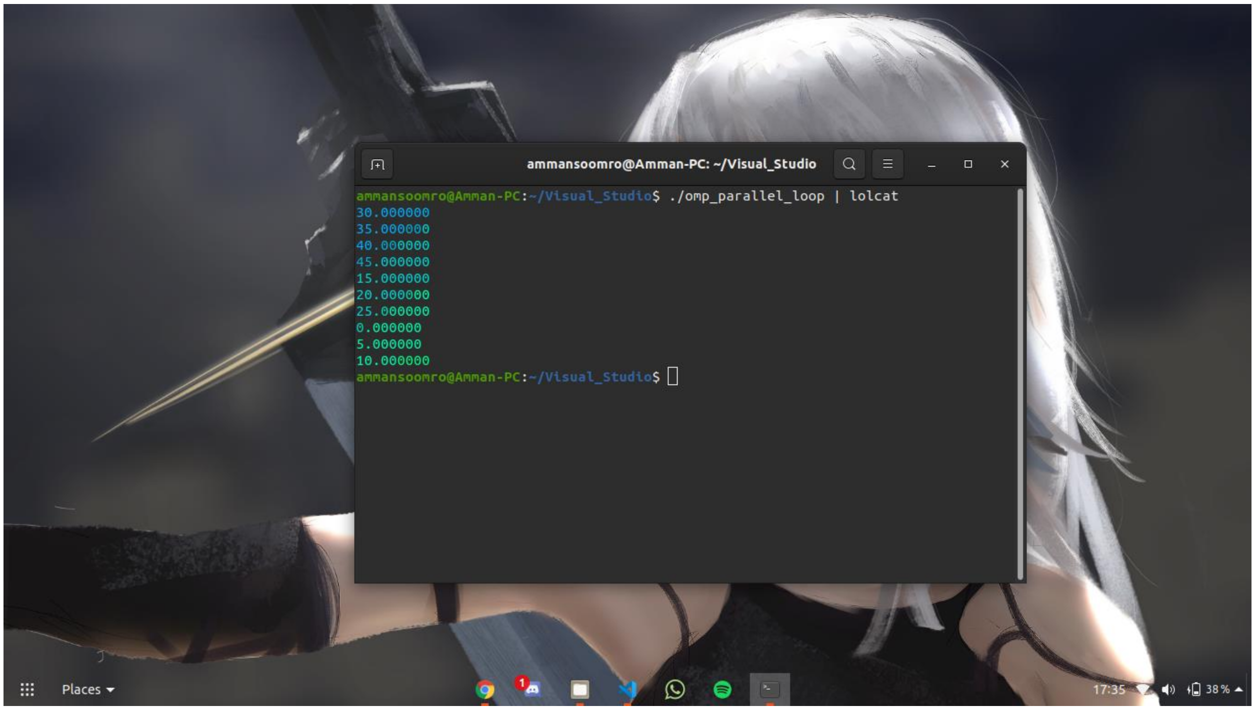
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp2 | lolcat
Hello World from thread 2
Hello World from thread 3
Hello World from thread 1
Hello World from thread 0
ammansoomro@Amman-PC:~/Visual_Studio$
```

OMP_SYNC



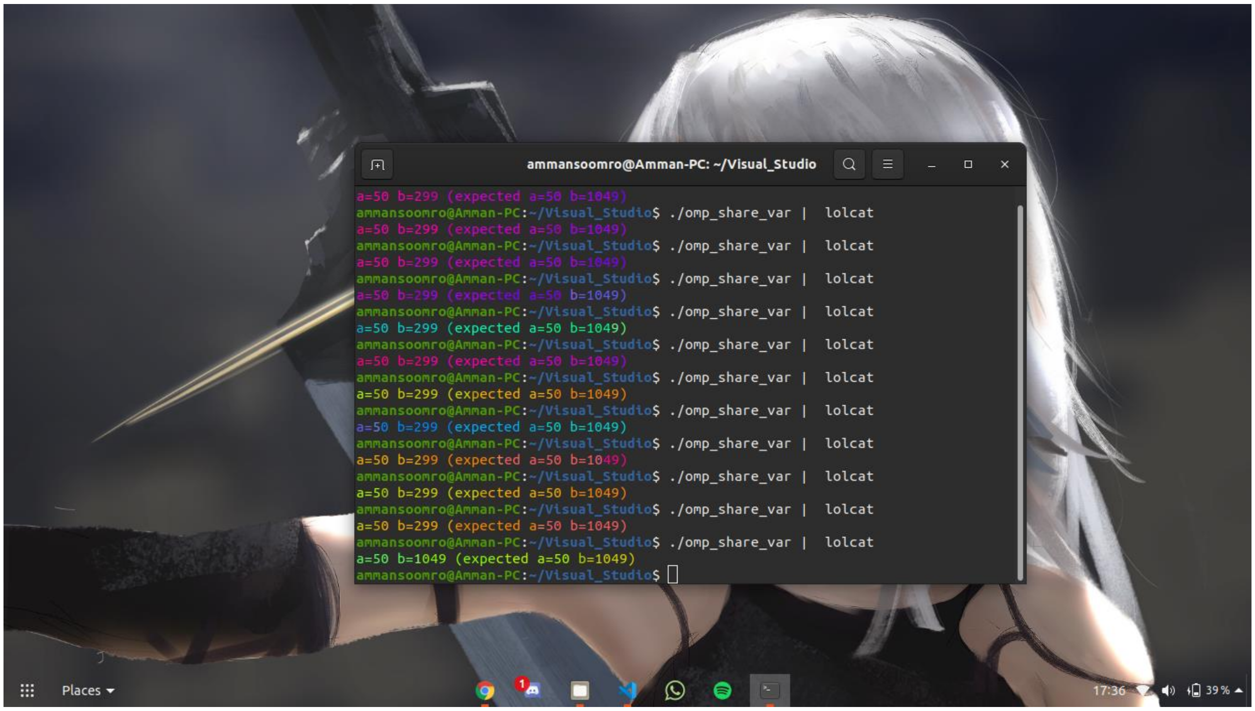
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_sync | lolcat
Hello World from thread 3
Hello World from thread 2
Hello World from thread 1
Hello World from thread 0
There are 4 threads
ammansoomro@Amman-PC:~/Visual_Studio$ 
```

OMP_PARALLEL_LOOP



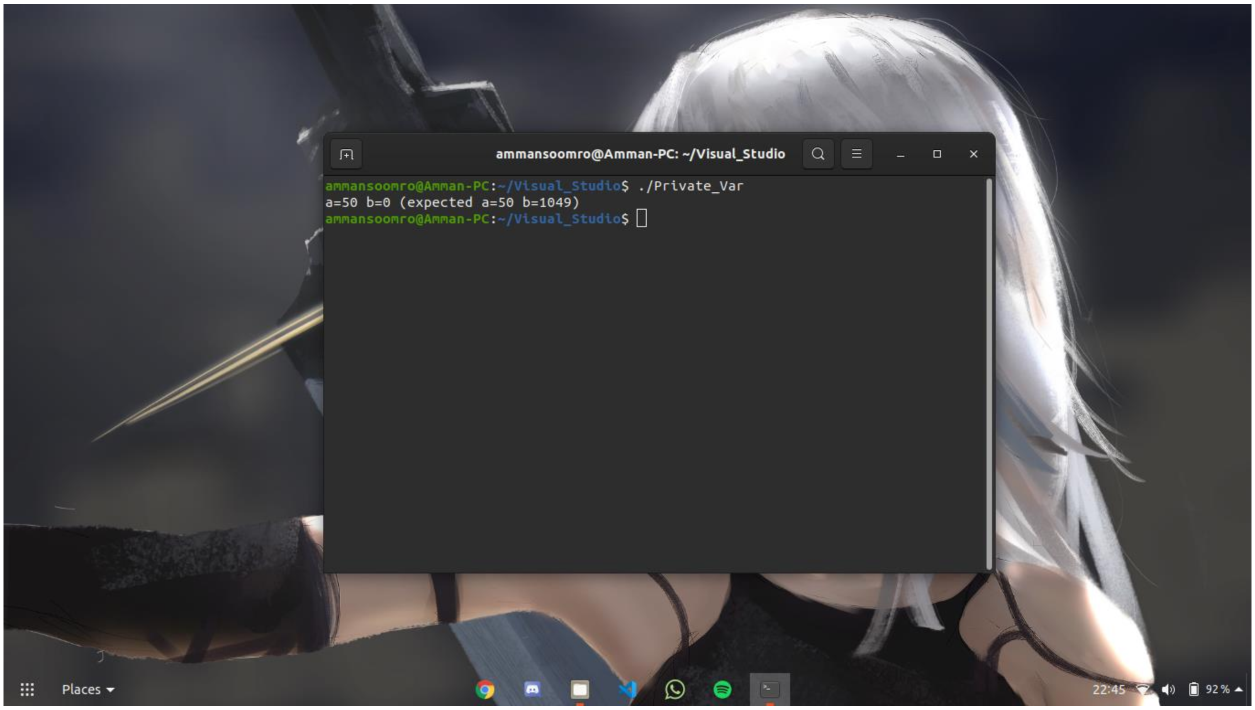
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_parallel_loop | lolcat
30.000000
35.000000
40.000000
45.000000
15.000000
20.000000
25.000000
0.000000
5.000000
10.000000
ammansoomro@Amman-PC:~/Visual_Studio$ 
```

OMP_SHARE_VAR



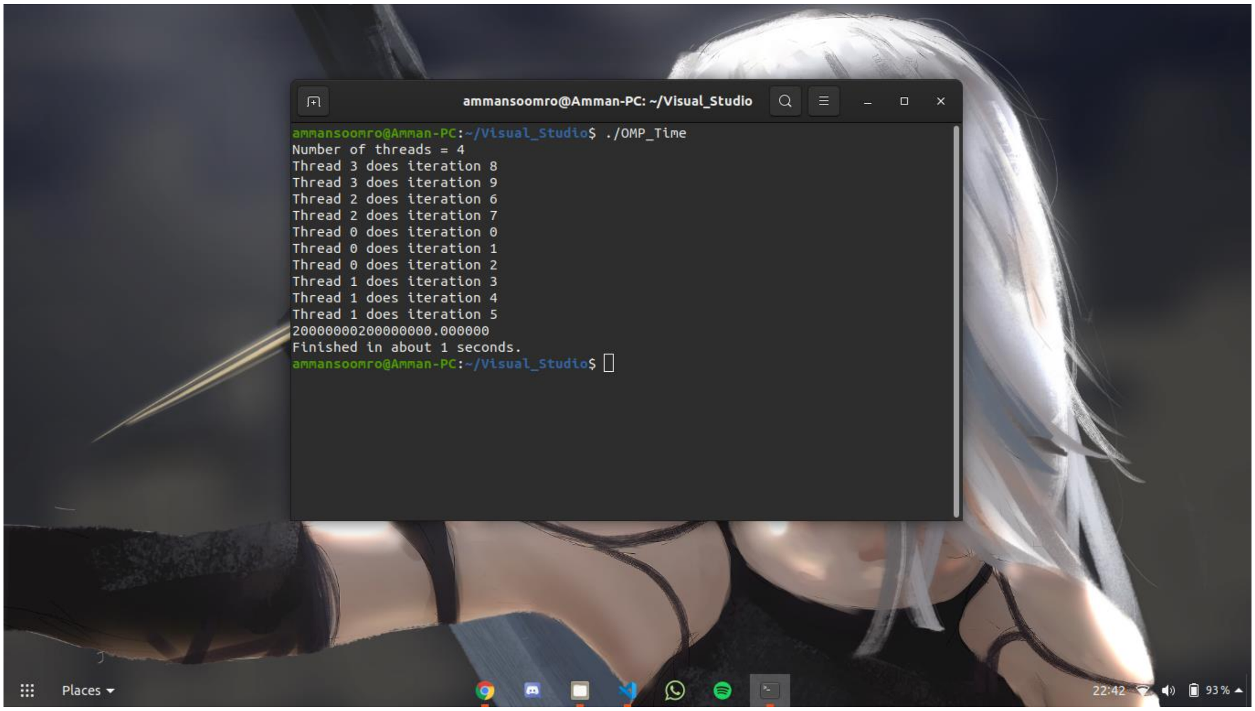
```
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=299 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$ ./omp_share_var |  lolcat
a=50 b=1049 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$
```

OMP_PRIVATE_VAR



```
ammansoomro@Amman-PC:~/Visual_Studio$ ./Private_Var
a=50 b=0 (expected a=50 b=1049)
ammansoomro@Amman-PC:~/Visual_Studio$
```

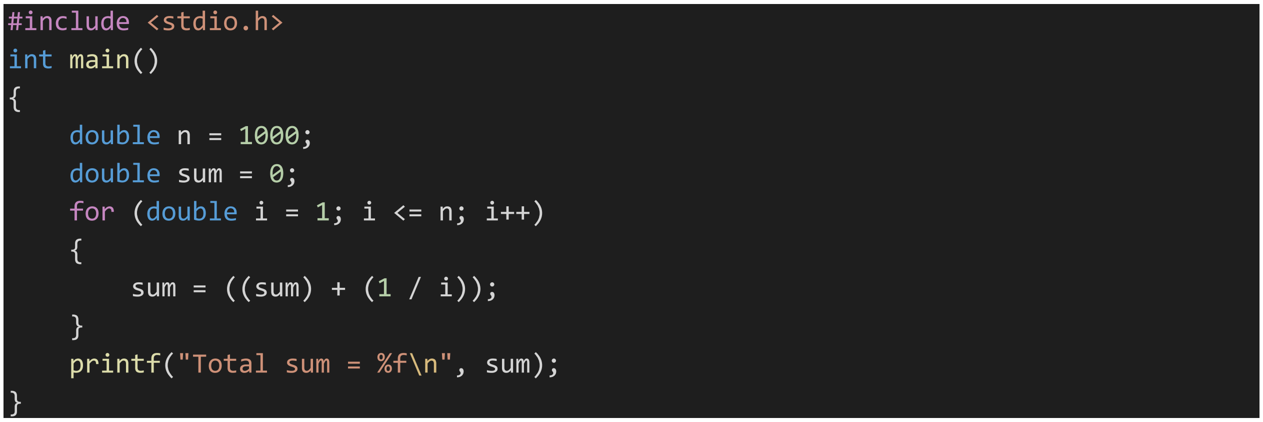
OMP_TIME



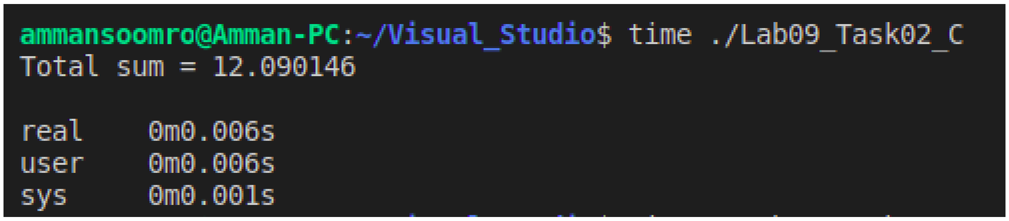
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./OMP_Time
Number of threads = 4
Thread 3 does iteration 8
Thread 3 does iteration 9
Thread 2 does iteration 6
Thread 2 does iteration 7
Thread 0 does iteration 0
Thread 0 does iteration 1
Thread 0 does iteration 2
Thread 1 does iteration 3
Thread 1 does iteration 4
Thread 1 does iteration 5
20000000200000000.000000
Finished in about 1 seconds.
ammansoomro@Amman-PC:~/Visual_Studio$ 
```

QUESTION NO 2

SUM USING C



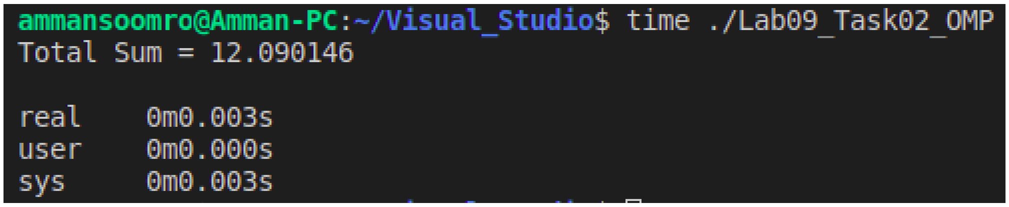
```c
#include <stdio.h>
int main()
{
    double n = 1000;
    double sum = 0;
    for (double i = 1; i <= n; i++)
    {
        sum = ((sum) + (1 / i));
    }
    printf("Total sum = %f\n", sum);
}
```



```
ammansoomro@Amman-PC:~/Visual_Studio$ time ./Lab09_Task02_C
Total sum = 12.090146

real    0m0.006s
user    0m0.006s
sys     0m0.001s
```

SUM USING OMP

```c
#include <stdio.h>
#include <omp.h>
#define N 1000
int main(void)
{
    double sum;
#pragma omp parallel default(shared)
    {
#pragma omp for
        for (int i = 1; i <= N; i++)
        {
            sum = ((sum) + (1/(double)i));
        }
    }
    printf("Total Sum = %f\n", sum);
}
```

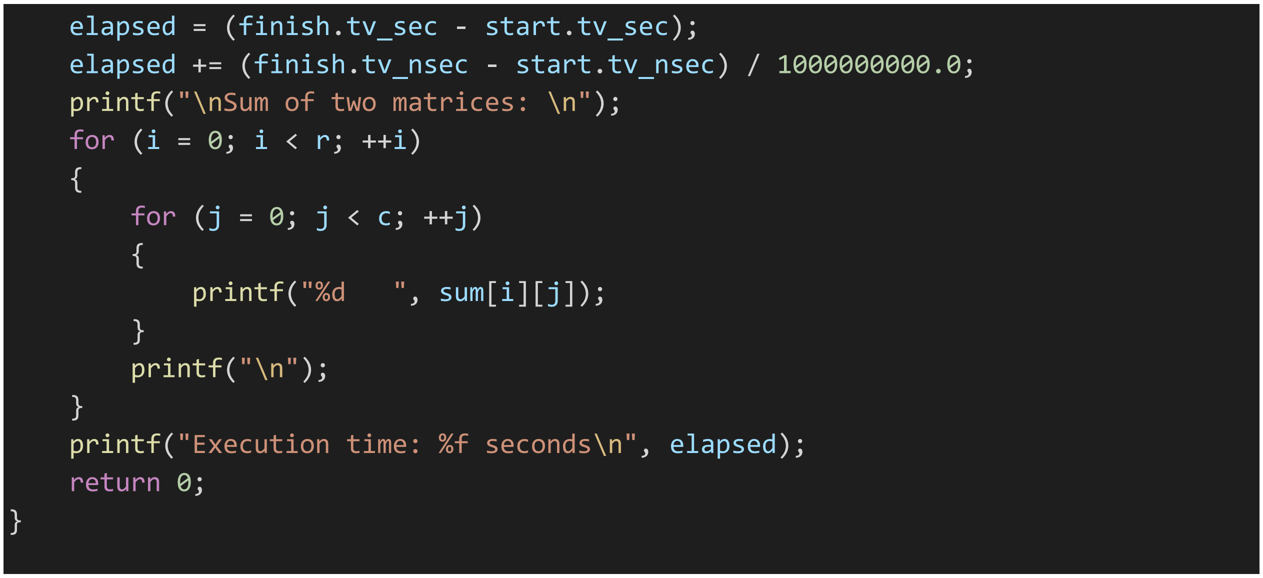


```
ammansoomro@Amman-PC:~/Visual_Studio$ time ./Lab09_Task02_OMP
Total Sum = 12.090146

real    0m0.003s
user    0m0.000s
sys     0m0.003s
```

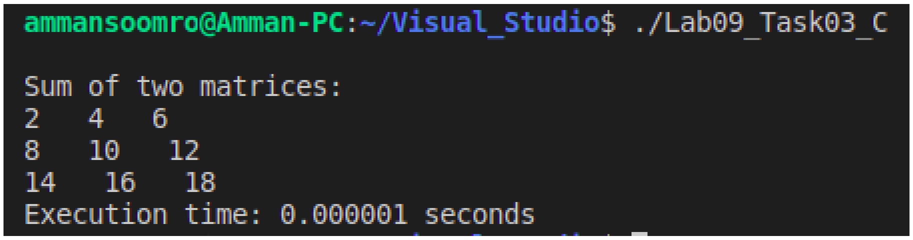
QUESTION NO 3

MATRIX SUM USING C

```c
#include <stdio.h>
#include <time.h>
int main()
{
    int r = 3, c = 3, a[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}, b[3][3] = {{1,
 2, 3}, {4, 5, 6}, {7, 8, 9}}, sum[3][3], i, j;
    struct timespec start, finish;
    double elapsed;
    clock_gettime(CLOCK_MONOTONIC, &start);
    for (i = 0; i < r; ++i)
    {
        for (j = 0; j < c; ++j)
        {
            sum[i][j] = a[i][j] + b[i][j];
        }
    }
    clock_gettime(CLOCK_MONOTONIC, &finish);
```

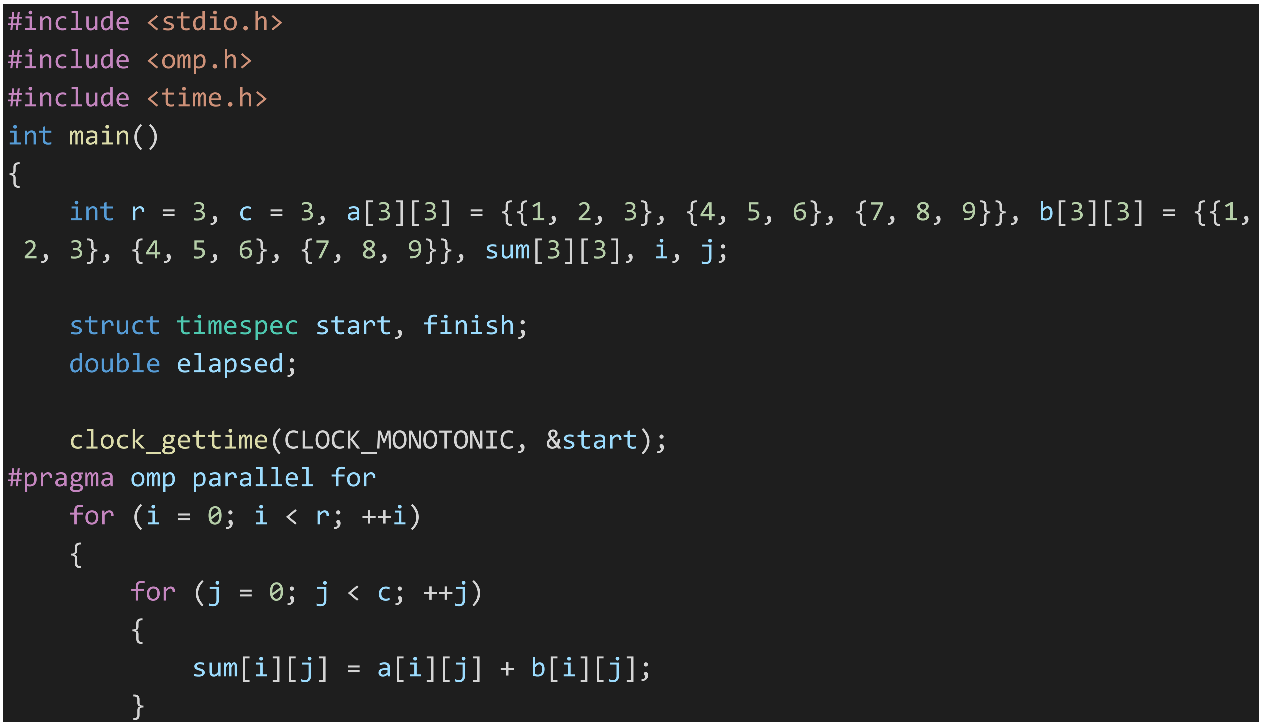


```
    elapsed = (finish.tv_sec - start.tv_sec);
    elapsed += (finish.tv_nsec - start.tv_nsec) / 1000000000.0;
    printf("\nSum of two matrices: \n");
    for (i = 0; i < r; ++i)
    {
        for (j = 0; j < c; ++j)
        {
            printf("%d   ", sum[i][j]);
        }
        printf("\n");
    }
    printf("Execution time: %f seconds\n", elapsed);
    return 0;
}
```

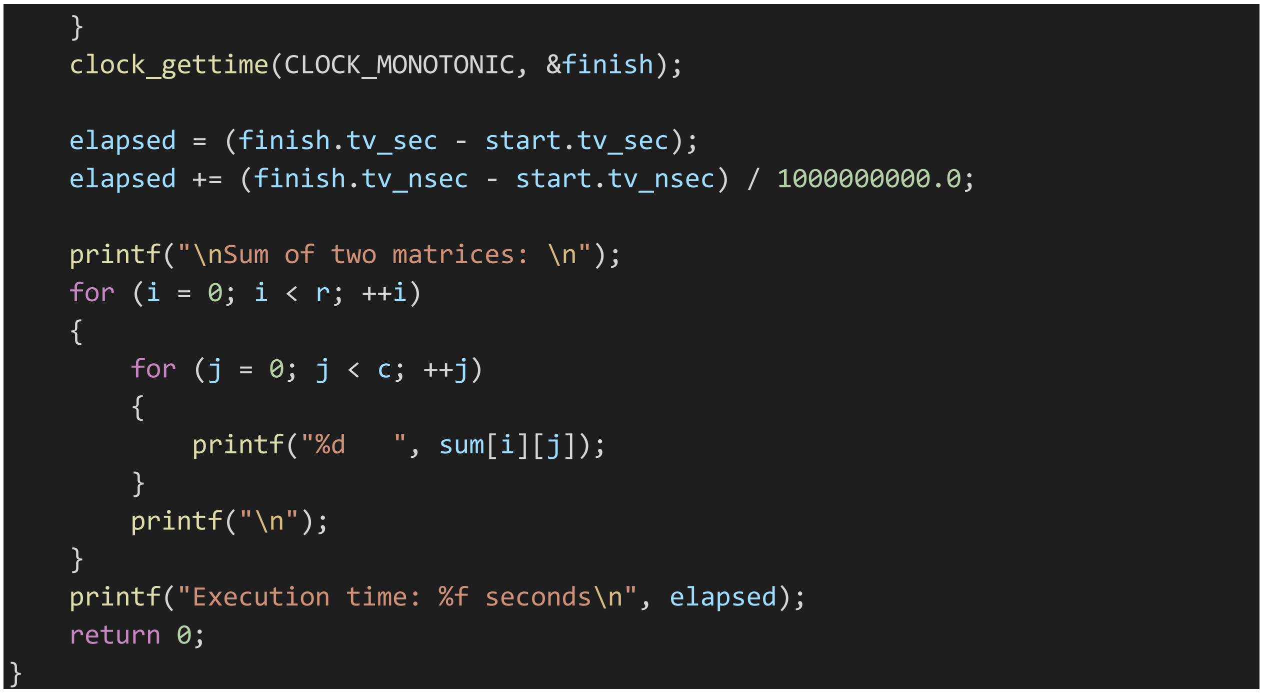


```
ammansoomro@Amman-PC:~/Visual_Studio$ ./Lab09_Task03_C

Sum of two matrices:
2   4   6
8   10   12
14   16   18
Execution time: 0.000001 seconds
```

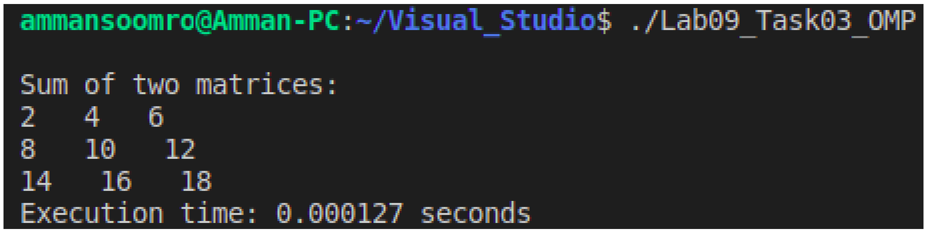
MATRIX SUM USING OMP



```
#include <stdio.h>
#include <omp.h>
#include <time.h>
int main()
{
    int r = 3, c = 3, a[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}, b[3][3] = {{1,
 2, 3}, {4, 5, 6}, {7, 8, 9}}, sum[3][3], i, j;

    struct timespec start, finish;
    double elapsed;

    clock_gettime(CLOCK_MONOTONIC, &start);
#pragma omp parallel for
    for (i = 0; i < r; ++i)
    {
        for (j = 0; j < c; ++j)
        {
            sum[i][j] = a[i][j] + b[i][j];
        }
```



```
    }
    clock_gettime(CLOCK_MONOTONIC, &finish);

    elapsed = (finish.tv_sec - start.tv_sec);
    elapsed += (finish.tv_nsec - start.tv_nsec) / 1000000000.0;

    printf("\nSum of two matrices: \n");
    for (i = 0; i < r; ++i)
    {
        for (j = 0; j < c; ++j)
        {
            printf("%d   ", sum[i][j]);
        }
        printf("\n");
    }
    printf("Execution time: %f seconds\n", elapsed);
    return 0;
}
```



```
ammansoomro@Amman-PC:~/Visual_Studio$ ./Lab09_Task03_OMP

Sum of two matrices:
2   4   6
8   10   12
14   16   18
Execution time: 0.000127 seconds
```