2. Write an OpenMP program that divides the Iterations into chunks containing 2 iterations, respectively (OMP_SCHEDULE=static,2). Its input should be the number of iterations, and its output should be which iterations of a parallelized for loop are executed by which thread.For example, if there are two threads and four iterations, the output might be the following:

a. Thread 0 : Iterations 0 –– 1

b. Thread 1 : Iterations 2 – 3

```
#include <stdio.h>

#include <omp.h>


int main(int argc, char* argv[]) {
    int n, i;


    if (argc != 2) {
        printf("Usage: %s <num_iterations>\n",
argv[0]);
```

```
        return 1;
    }

    n = atoi(argv[1]);   // number of iterations

    // Parallel region
    #pragma omp parallel private(i)
    {
        #pragma omp for schedule(static,2)
        for (i = 0; i < n; i++) {
            int tid = omp_get_thread_num();

            // Print which iteration belongs to which
thread
            printf("Thread %d executes iteration
%d\n", tid, i);
        }
    }
    return 0;
}
```

Output:

```
export OMP_NUM_THREADS=2
./a.out 4

Thread 0 executes iteration 0
Thread 0 executes iteration 1
Thread 1 executes iteration 2
Thread 1 executes iteration 3
```