###### 12b

#include <stdio.h>

#include <omp.h>

int isPrime(int num) {

if (num <= 1)

return 0;

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0)

return 0;

}

return 1;

}

void printPrimes() {

#pragma omp parallel for

for (int i = 2; i <= 100; i++) {

if (isPrime(i)) {

#pragma omp critical

printf("%d ", i);

}

}

}

void printOddNumbers() {

#pragma omp parallel for

for (int i = 1; i <= 100; i += 2) {

#pragma omp critical

printf("%d ", i);

}

}

void printFibonacci() {

int n = 100;

int a = 0, b = 1, nextTerm;

#pragma omp parallel

{

#pragma omp single

{

for (int i = 0; i < n; i++) {

#pragma omp task

{

#pragma omp critical

printf("%d ", a);

nextTerm = a + b;

a = b;

b = nextTerm;

}

}

}

}

}

int main() {

#pragma omp parallel sections

{

#pragma omp section

{

printf("Prime numbers:\n");

printPrimes();

printf("\n");

}

#pragma omp section

{

printf("Odd numbers:\n");

printOddNumbers();

printf("\n");

}

#pragma omp section

{

printf("Fibonacci series:\n");

printFibonacci();

printf("\n");

}

}

return 0;

}

Open a text editor and save the code into a file, for example, openmp\_threads.c.

* Open a terminal and navigate to the directory where you saved the file.
* Install the OpenMP library by running the following command in the terminal:

arduino

* sudo apt-get install libomp-dev
* Compile the program using the gcc compiler with OpenMP support. Run the following command in the terminal:
* gcc -fopenmp openmp\_threads.c -o openmp\_threads
* After successful compilation, you can execute the program by running:

bash

* ./openmp\_threads
* The program will start running, and you'll see the output in the terminal, which includes the prime numbers, odd numbers, and the Fibonacci series.