Parallel Computing (BCS702)

Program - 3

Write an openMP program to calculate n Fibonacci numbers using tasks. This demonstrates task-based parallelism and recursive task creation.

```
#include <stdio.h>
#include <omp.h>


// Recursive Fibonacci with OpenMP tasks
long long fib(int n) {
    long long x, y;

    if (n <= 1) return n;

    #pragma omp task shared(x)
    x = fib(n - 1);

    #pragma omp task shared(y)
    y = fib(n - 2);

    #pragma omp taskwait
    return x + y;
}

int main() {
    int n;
    printf("Enter number of Fibonacci terms: ");
    scanf("%d", &n);

    double start = omp_get_wtime();

    #pragma omp parallel
    {
        #pragma omp single
        {
            for (int i = 0; i < n; i++) {
                long long res = fib(i);
                printf("Fib(%d) = %lld\n", i, res);
            }
```

```
        }
    }

    double end = omp_get_wtime();
    printf("Time taken: %f seconds\n", end - start);

    return 0;
}
```

Output :

Enter number of Fibonacci terms: 7

Fib(0) = 0
Fib(1) = 1
Fib(2) = 1
Fib(3) = 2
Fib(4) = 3
Fib(5) = 5
Fib(6) = 8
Time taken: 0.002345 seconds