

Assign 3

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

#define N 100
#define NUM_PROCESSORS 4

int main()
{
    int arr[N];
    for (int i = 0; i < N; i++)
    {
        arr[i] = sizeof(int) * i;
    }

    int sum = 0;
    int PARTIAL_SUM[NUM_PROCESSORS];

#pragma omp parallel num_threads(NUM_PROCESSORS)
    {
        int thread_id = omp_get_thread_num();
        int start = thread_id * (N / NUM_PROCESSORS);
        int end = (thread_id + 1) * (N / NUM_PROCESSORS);
        PARTIAL_SUM[thread_id] = 0;

        for (int i = start; i < end; i++)
        {
            PARTIAL_SUM[thread_id] += arr[i];
        }
    }

    for (int i = 0; i < NUM_PROCESSORS; i++)
    {
        sum += PARTIAL_SUM[i];
        printf("Partial sum of thread %d: %d\n", i, PARTIAL_SUM[i]);
    }

    printf("Sum: %d\n", sum);

    return 0;
}
```

```
PS D:\Acad\DS Assign\Assign3> gcc -fopenmp main.c -o output
PS D:\Acad\DS Assign\Assign3> .\output
Partial sum of thread 0: 1200
Partial sum of thread 1: 3700
Partial sum of thread 2: 6200
Partial sum of thread 3: 8700
Sum: 19800
PS D:\Acad\DS Assign\Assign3> █
```