

## main.c

```
1  #include<stdio.h>
2  #include<omp.h>
3
4  #define N 100
5  #define NUM_PROCESSORS 4
6
7
8  int main()
9  {
10     int arr[N];
11     for (int i = 0; i < N; i++)
12     {
13         arr[i] = sizeof(int) * i;
14     }
15
16
17     int sum = 0;
18     int PARTIAL_SUM[NUM_PROCESSORS];
19
20
21     #pragma omp parallel num_threads(NUM_PROCESSORS)
22     {
23         int thread_id = omp_get_thread_num();
24         int start = thread_id * (N / NUM_PROCESSORS);
25         int end = (thread_id + 1) * (N / NUM_PROCESSORS);
26         PARTIAL_SUM[thread_id] = 0;
27
28
29         for (int i = start; i < end; i++)
30         {
31             PARTIAL_SUM[thread_id] += arr[i];
32         }
33     }
34
35
36     for (int i = 0; i < NUM_PROCESSORS; i++)
37     {
38         sum += PARTIAL_SUM[i];
39         printf("Partial sum of thread %d: %d\n", i, PARTIAL_SUM[i]);
40     }
41
42
43     printf("Sum: %d\n", sum);
44
45
46     return 0;
47 }
48
```