

```
1  // prodotto scalare tra 2 vettori
2
3  #include <omp.h>
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  #define N 100
8
9  int main (int argc, char *argv[]) {
10
11     double a[N], b[N];
12     double localsum, sum = 0.0;
13     int i, tid, nthreads;
14
15     #pragma omp parallel shared(a,b,sum) private(i, localsum)
16     {
17         /* Get thread number */
18         tid = omp_get_thread_num();
19
20         /* Only master thread does this */
21         if (tid == 0) {
22             nthreads = omp_get_num_threads();
23             printf("Number of threads = %d\n", nthreads);
24         }
25
26         /* Initialization */
27         #pragma omp for
28         for (i=0; i < N; i++)
29             a[i] = b[i] = (double)i;
30
31         localsum = 0.0;
32
33         /* Compute the local sums of all products */
34         #pragma omp for
35         for (i=0; i < N; i++)
36             localsum = localsum + (a[i] * b[i]);
37
38         #pragma omp critical
39             sum = sum+localsum;
40
41     } /* End of parallel region */
42
43     printf("    Sum = %2.1f\n",sum);
44     exit(0);
45 }
46
```