

```

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

int main(int argc, char* argv[]) {
    #pragma omp parallel
    {
        printf("Hello, World! from thread %d\n",
omp_get_thread_num());
    }
    return 0;
}

```

```

> gcc hello_world.c -fopenmp -o hello_world
> ./hello_world
Hello, World! from thread 7
Hello, World! from thread 5
Hello, World! from thread 2
Hello, World! from thread 4
Hello, World! from thread 6
Hello, World! from thread 3
Hello, World! from thread 0
Hello, World! from thread 1

```

```

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

long fib(int n) {
    return (n < 2?1:fib(n-1) + fib(n-2));
}

```

```

int main(int argc, char* argv[]) {
    int n = 45;
    #pragma omp parallel
    {
        int t = omp_get_thread_num();
        printf("%d: %ld\n", t, fib(n+t));
    }
    return 0;
}

```

```

> gcc fibonacci.c -fopenmp -o fibonacci
> ./fibonacci
0: 1836311903
1: 2971215073
2: 4807526976
3: 7778742049

```

```
4: 12586269025
5: 20365011074
6: 32951280099
7: 53316291173
```

```
> OMP_NUM_THREADS=1 /usr/bin/time ./fibonacci
0: 14930352
0.24user 0.00system 0:00.24elapsed 100%CPU
```

```
> OMP_NUM_THREADS=2 /usr/bin/time ./fibonacci
0: 14930352
1: 24157817
0.74user 0.00system 0:00.43elapsed 173%CPU
```

```
> OMP_NUM_THREADS=3 /usr/bin/time ./fibonacci
0: 14930352
1: 24157817
2: 39088169
1.52user 0.00system 0:00.72elapsed 210%CPU
```

```
> OMP_NUM_THREADS=4 /usr/bin/time ./fibonacci
0: 14930352
1: 24157817
2: 39088169
3: 63245986
2.59user 0.00system 0:01.09elapsed 237%CPU
```

```
> OMP_NUM_THREADS=5 /usr/bin/time ./fibonacci
0: 14930352
1: 24157817
2: 39088169
3: 63245986
4: 102334155
4.54user 0.00system 0:01.70elapsed 266%CPU
```

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

```
int main(int argc, char* argv[]) {
    if(argc < 2) {
        printf("Not enough arguments.\n");
        return 0;
    }

    int n = atoi(argv[1]);
    if(n == 0) {
        printf("Invalid integer value.\n");
    }
}
```

```
    }  
  
    #pragma omp parallel for  
    for(int i=0; i<n; ++i) {  
        printf("%d: %d\n", i, omp_get_thread_num());  
    }  
    return 0;  
}
```

```
> gcc for_loop.c -fopenmp -o for_loop
```

```
> ./for_loop 15
```

```
0: 0
```

```
2: 1
```

```
3: 1
```

```
6: 3
```

```
4: 2
```

```
5: 2
```

```
8: 4
```

```
14: 7
```

```
7: 3
```

```
10: 5
```

```
11: 5
```

```
1: 0
```

```
12: 6
```

```
13: 6
```

```
9: 4
```