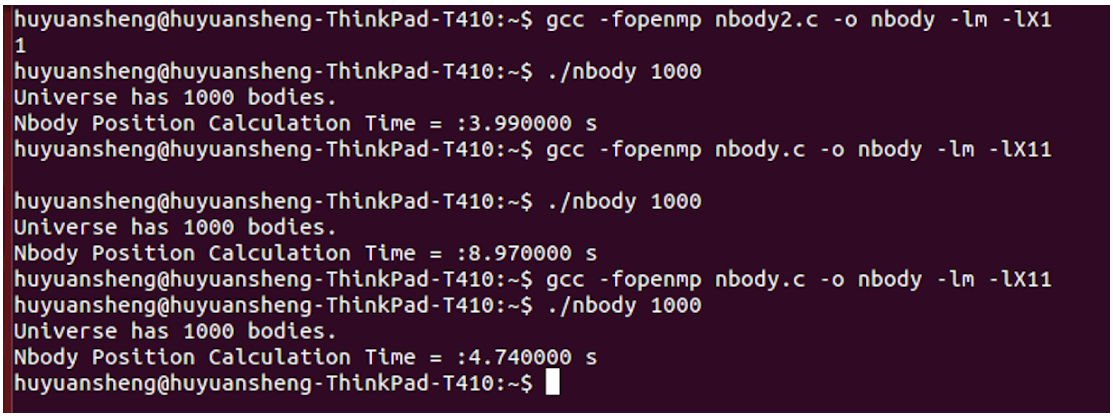
1) I parallelize the core function by add the pragma for before the two for loops in nbody.c , And I use the default schedule(block schedule)

2) For optimization in question two, I use a simplify algorithm in nbody2.c, and there are divided in four loops and add four pragma, the usage of the new core function method is just like the textbook one, and the performance is much better than the nbody.c (see the text book for reference)

3) The test case is having 4 threads



```
huyuansheng@huyuansheng-ThinkPad-T410:~$ gcc -fopenmp nbody2.c -o nbody -lm -lX1
1
huyuansheng@huyuansheng-ThinkPad-T410:~$ ./nbody 1000
Universe has 1000 bodies.
Nbody Position Calculation Time = :3.990000 s
huyuansheng@huyuansheng-ThinkPad-T410:~$ gcc -fopenmp nbody.c -o nbody -lm -lX11

huyuansheng@huyuansheng-ThinkPad-T410:~$ ./nbody 1000
Universe has 1000 bodies.
Nbody Position Calculation Time = :8.970000 s
huyuansheng@huyuansheng-ThinkPad-T410:~$ gcc -fopenmp nbody.c -o nbody -lm -lX11
huyuansheng@huyuansheng-ThinkPad-T410:~$ ./nbody 1000
Universe has 1000 bodies.
Nbody Position Calculation Time = :4.740000 s
huyuansheng@huyuansheng-ThinkPad-T410:~$ 
```

the first one is optimization parallelize

the second one is the serial one (by comments the pragma statment)

the third one is the naïve paralleize one