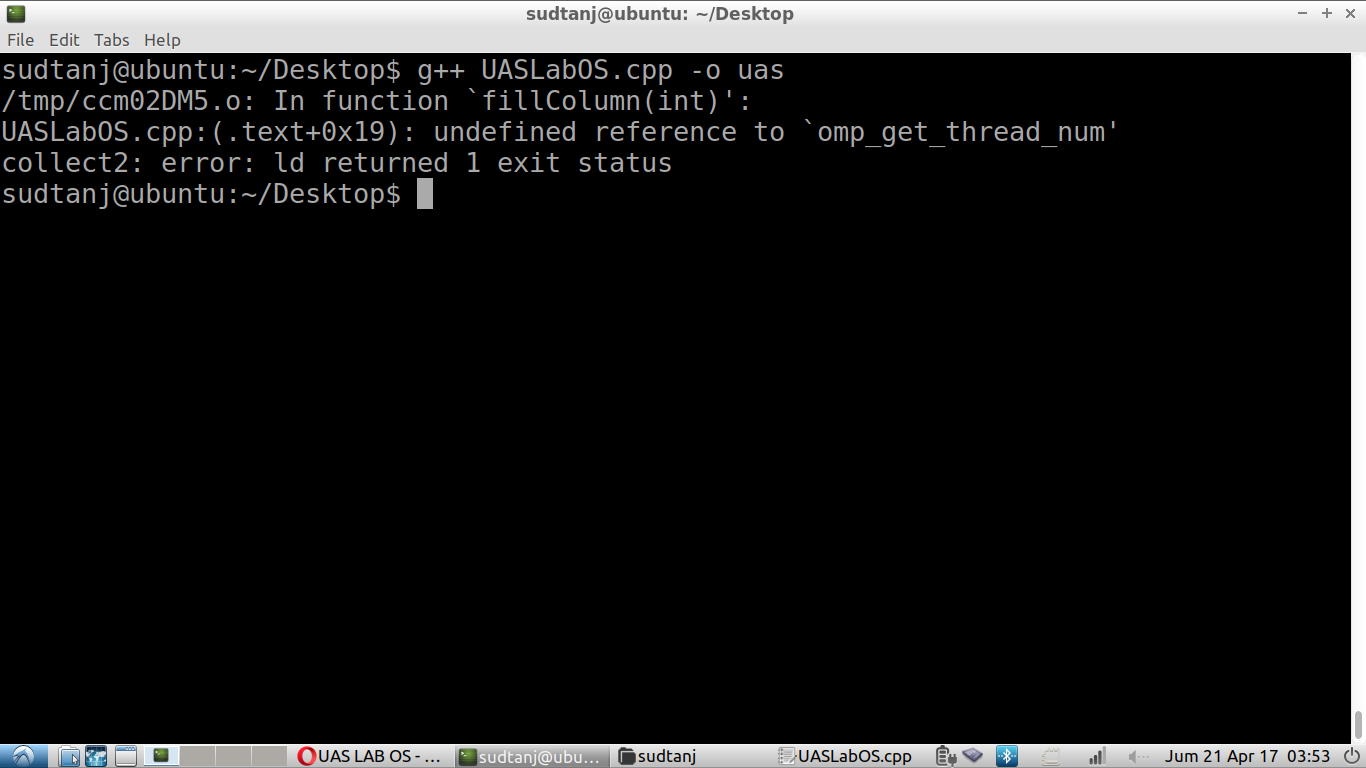
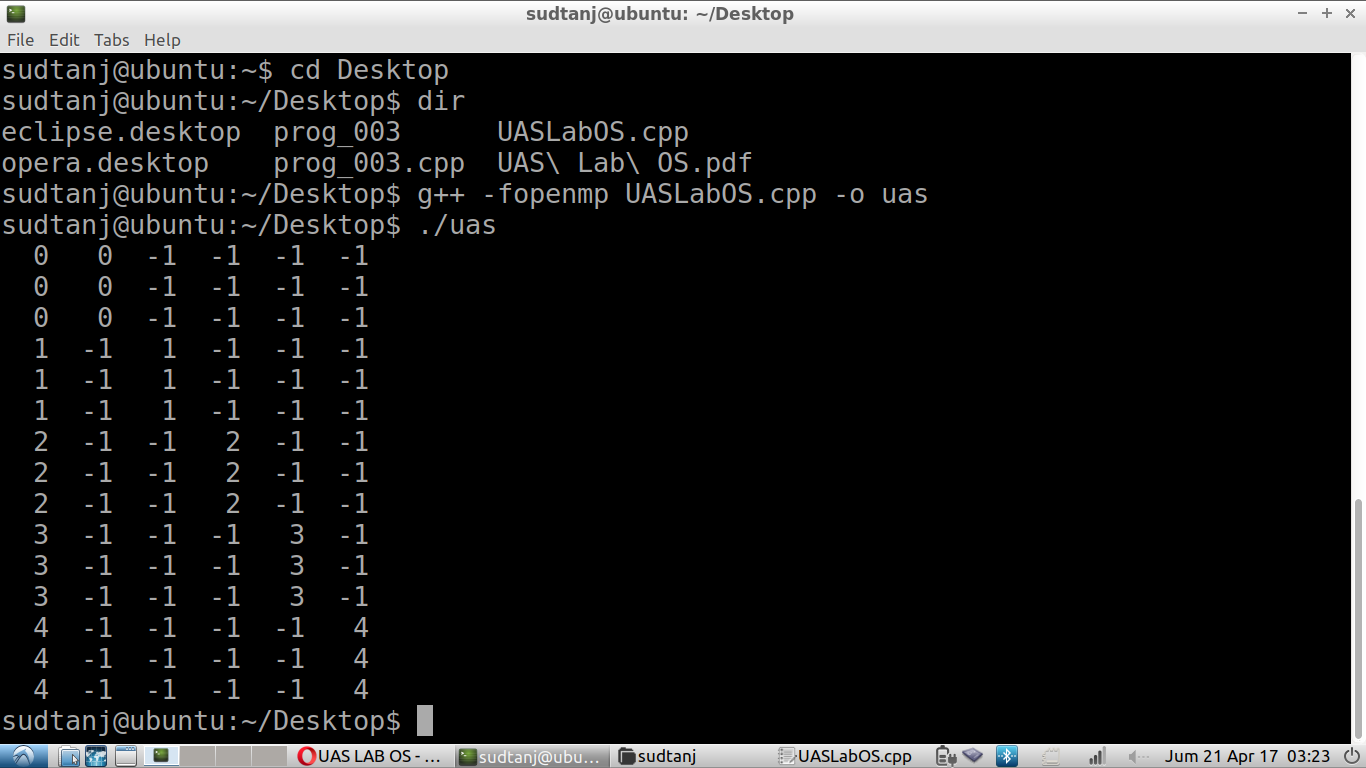
1. Explain the output of this program if it is compiled without –fopenmp flag!



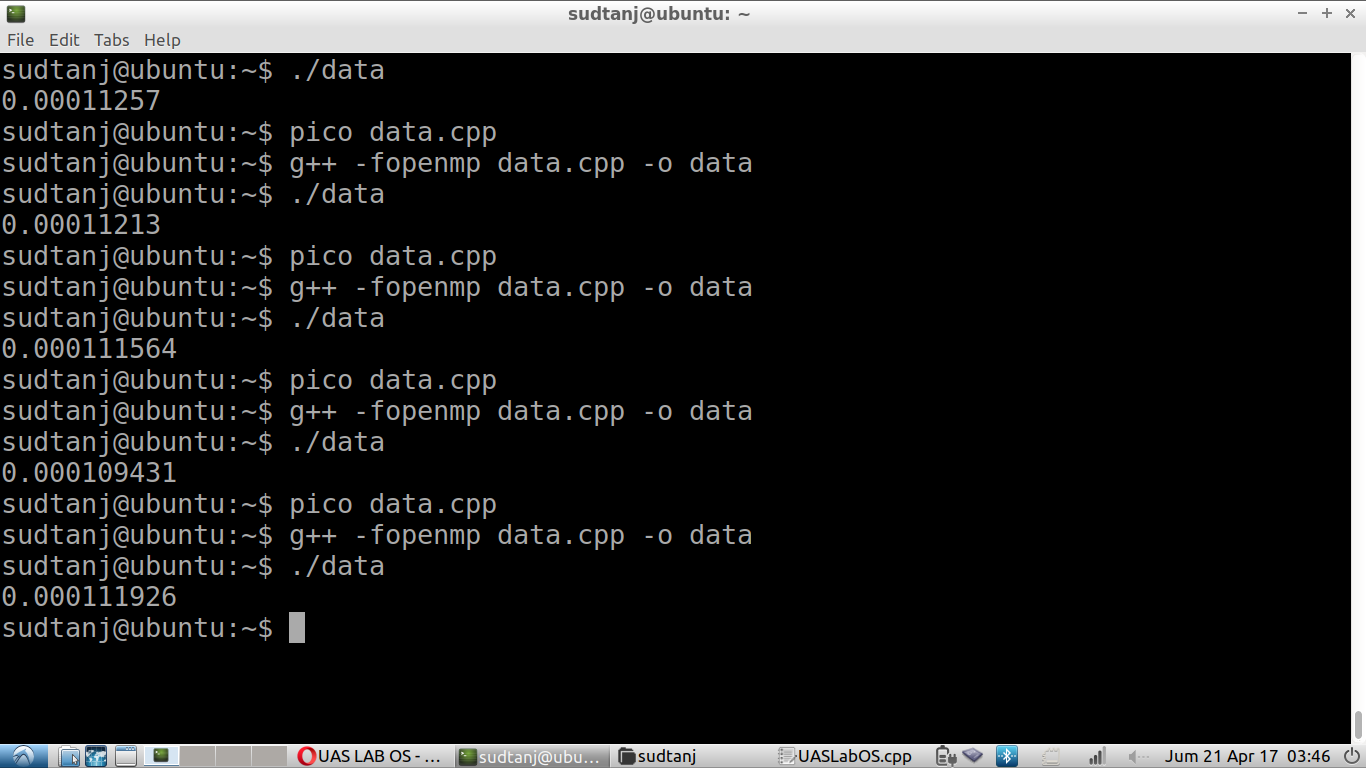
* 1. Answer = The error reference is occurred because the library is not included when we compiling the cpp file. The compiler g++ in this case doesn’t know what is omp\_get\_num\_thread() function because that function is only defined inside omp.h library where omp.h library need -fopenmp parameter in the compiler to be included when the assembly is happening.

1. Explain carefully the output of this program with –fopenmp flag, also why the output is different than the serial version of this program!

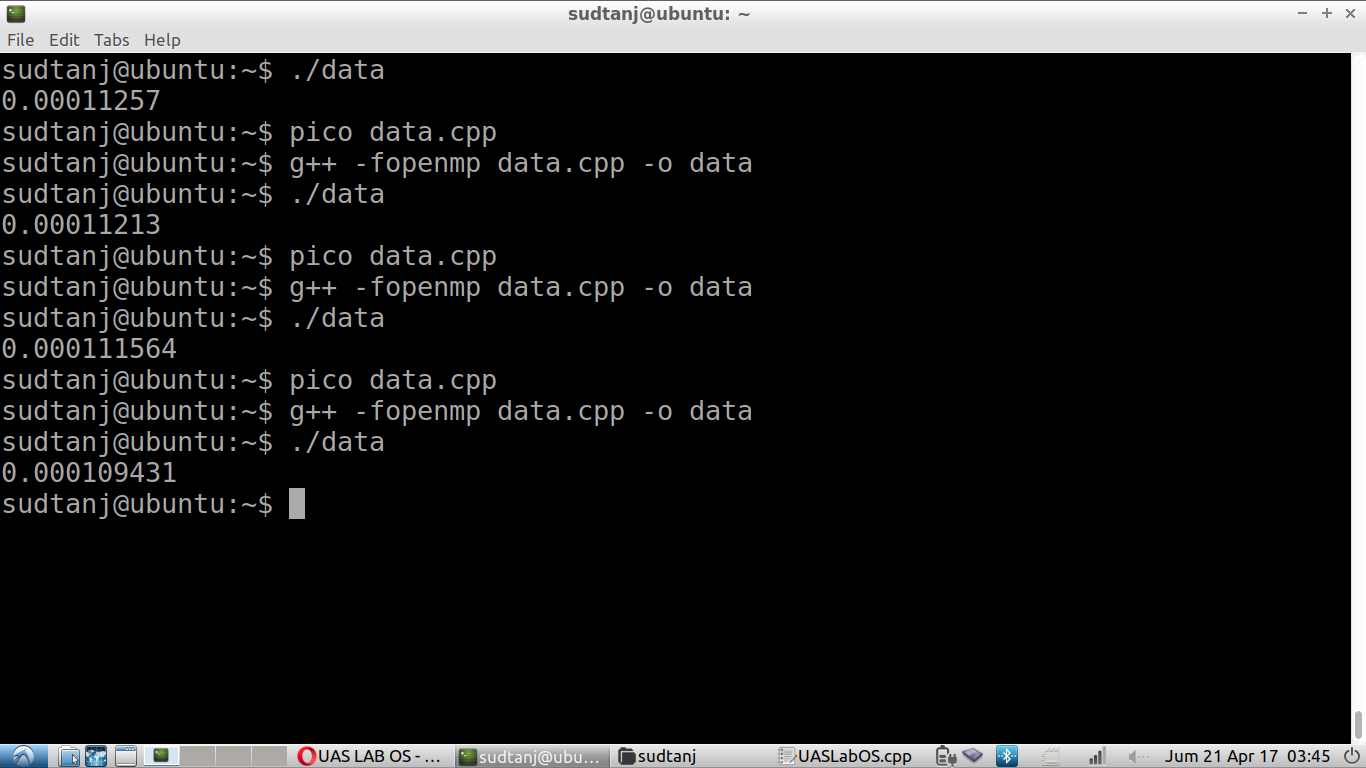


* 1. Answer = The output is different from the serial version because in the serial version, all thread will be access column 0 known in the code “fillColumn(0); “ because of this all array in the column 0 will be filled with the thread number meanwhile in the other column the column 1 is filled with thread 0 when the other thread that is accessing column 2-5 is still -1 because the main thread that is executing still thread 0, after 3 times executed, it moved to thread 1 and thread 0 is suspended. Thread 1 will execute column 2 and filled the number with it thread number which is 1 in row 3 - 6 and so on until the for loop of the all thread is finished.

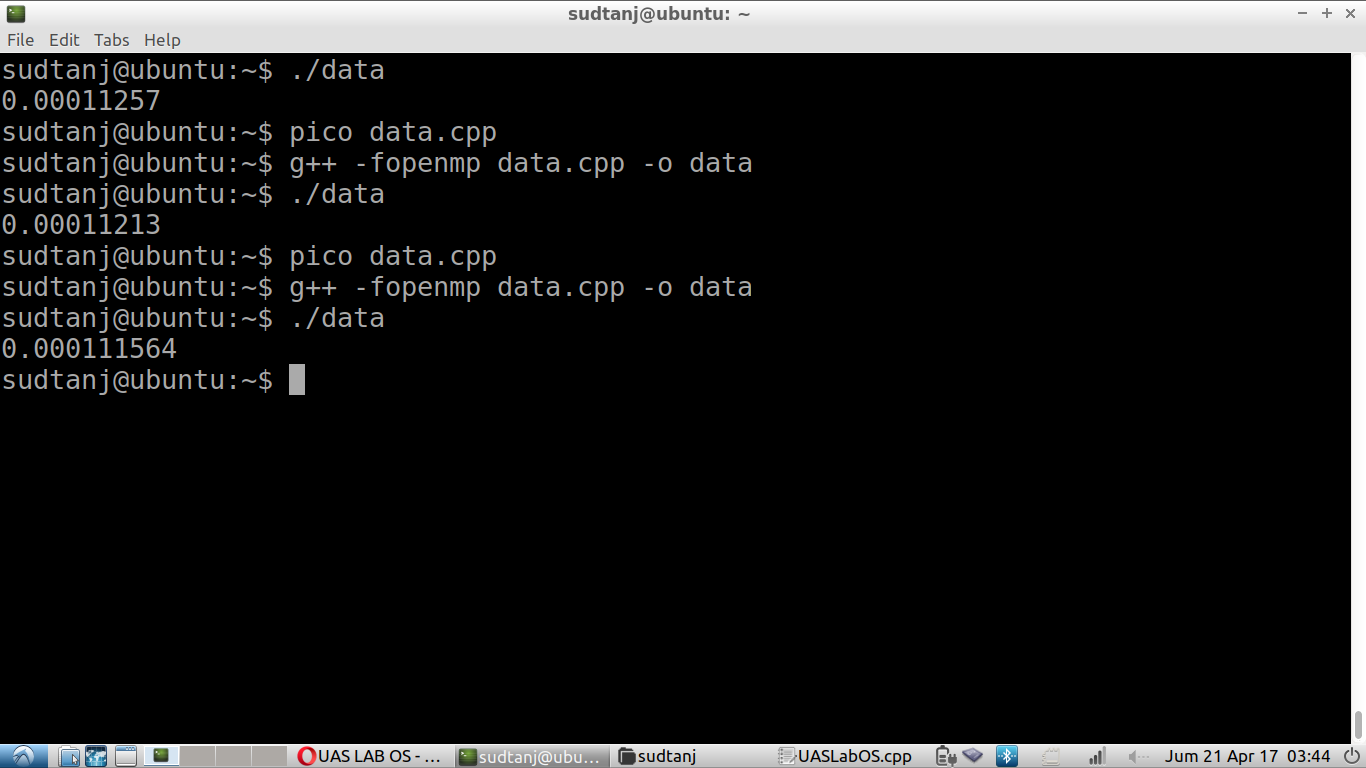
1. Suppose that we have a computer with 4 cores. Suppose we use OpenMP to parallelize a for-loop that initializes to zero the upper triangle of a 100 × 100 matrix.
   1. Answer =
      1. schedule(static) takes 0.000109431 in thread time



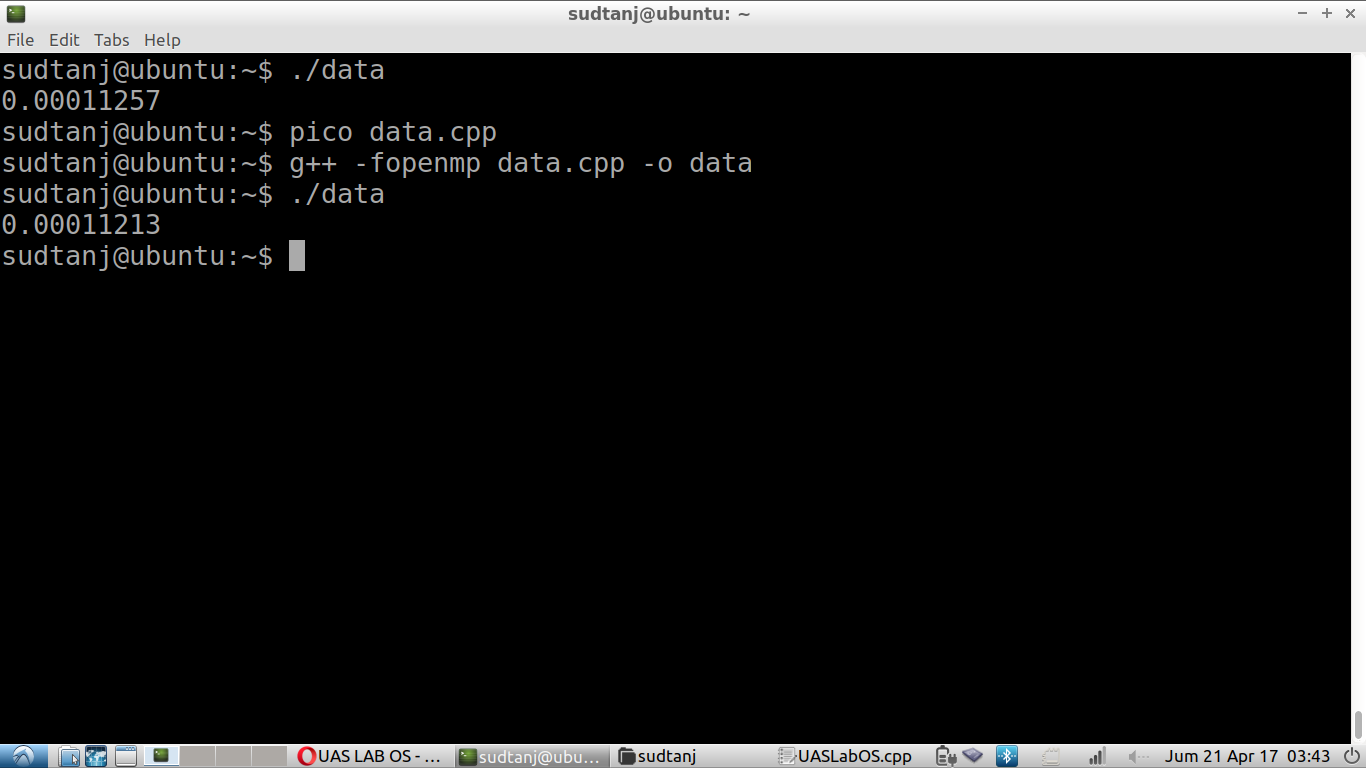
* + 1. schedule(static,10) does 0.000109431 in thread time



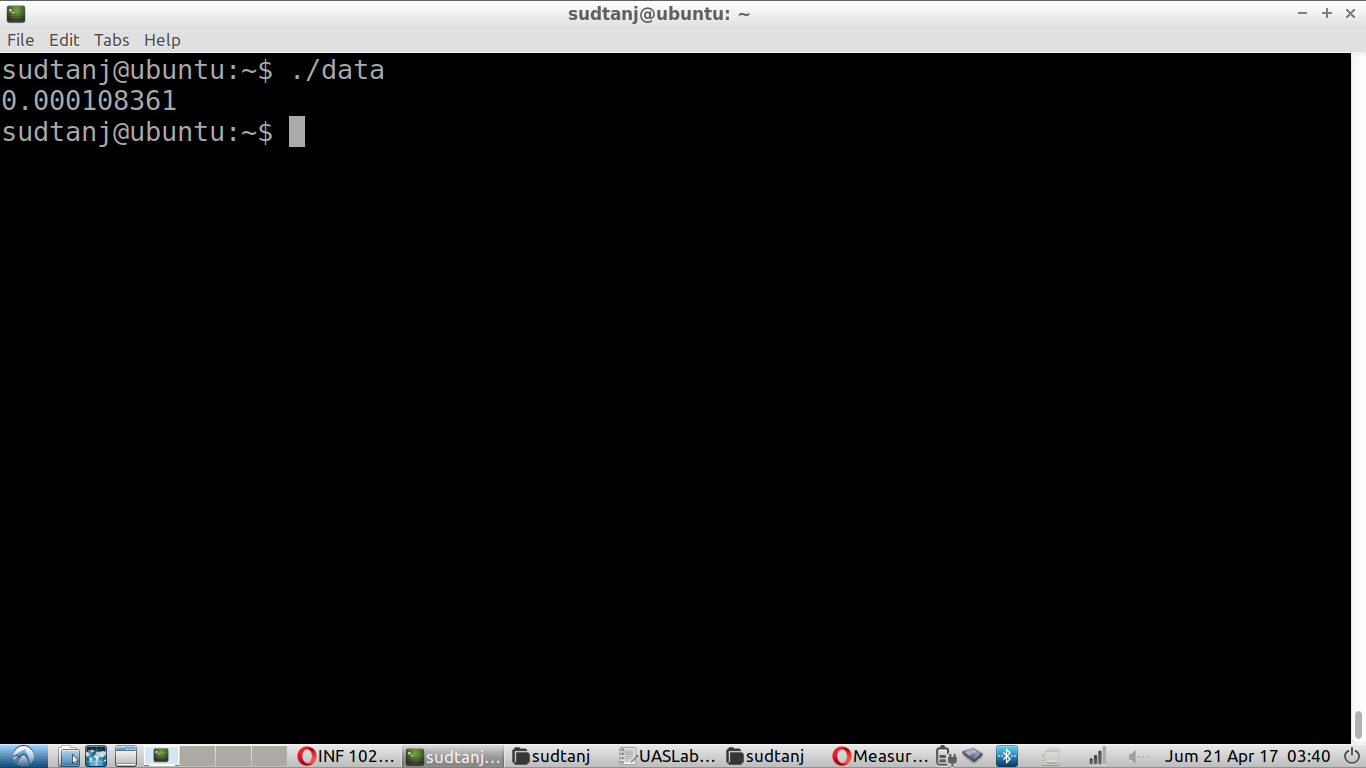
* + 1. Schedule(static, 1) takes 0.000111564 in thread time



* + 1. schedule(dynamic,1) takes 0.00011213 in thread time



* + 1. schedule(dynamic, 10) takes 0.000108361 in thread time



* + 1. Schedule (dynamic,20) takes 0.00011257 in thread time to do the task

