

Write an OpenMP parallel program to find the number of pythagorean triplets ¹ (a, b, c) where $a < b \leq 50000$. Avoid printing out each of the pythagorean triplets; only count them. Printing is inherently sequential, and doing so will take a large percentage of the total elapsed time.

```
#include <iostream>    // Sample sequential C++ program.
#include <cmath>
#include "etime.h"
using namespace std ;

const int N = 50000 ;
int main()
{
    int a, b, c, t, count ;

    count = 0 ;
    tic() ; // Start timer
    for ( a = 1 ; a <= N ; a++ ) {
        for ( b = a+1 ; b <= N ; b++ ) {
            t = a*a + b*b ;
            c = round( sqrt( (double) t )) ;
            if ( t == (c*c) ) count++ ;
        }
    }
    toc() ; // Stop timer
    cout<< "count = " << count << endl ;
    cout<< "elapsed time = " << etime() << endl ;
}
```

A simple modification of the program above yields the following times and speedups:

Threads	1	2	3	4	5	6	7	8	9	10
Times	16.3	8.07	5.55	4.12	3.32	2.81	2.42	2.43	2.11	1.73

Threads	11	12	13	14	15	16	17	18	19	20
Times	1.80	1.53	1.55	1.43	1.36	1.25	1.20	1.12	1.06	1.01

Threads	1	2	3	4	5	6	7	8	9	10
Speedup	1.00	2.02	2.94	3.95	4.90	5.79	6.74	6.71	7.72	9.42

Threads	11	12	13	14	15	16	17	18	19	20
Speedup	9.06	10.7	10.5	11.4	12.0	13.0	13.6	14.6	15.3	16.2

¹I.e., integers a, b , and c such that $a^2 + b^2 = c^2$.

