```c
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "common.h"


int main(int argc, char** argv)
{
    //value of sum as n->infinity
    double exactsum=pow((4.0*atan(1.0)),2)/6.0;

    //initialize some variables
    double sum=0;
    int lastN=0;

    //make the vector
    Vector v = createVector(pow(2,14));

    //divide the vector filling operation into each 2^k piece
    //in order to parallelize filling task while avoiding calculating
    //vector length(k) times
    for(int k=4;k<15;++k){

        //reset the sum after each 2^k is reached to avoid double
        //counting previously summed numbers
        double isum=0;

        //update number of iterations to complete next section of vector
        int nextN=pow(2,k);
        #pragma omp parallel for schedule(static) reduction(+:isum)
        for (int i=lastN;i<nextN;++i) {
            v->data[i] = 1.0/((double)(i+1)*(double)(i+1));

            //calculate the sum on the fly - saves a second for loop
            isum += v->data[i];
        }
        //add sum from current section of vector to total sum
        sum += isum;
        printf("difference at i=2^%2i: %1.16f\n", k, exactsum-sum);

        //update starting point for next section of vector
        lastN=nextN;
    }
}
```