```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
#include<time.h>
#include<math.h>
#include<float.h>
#define NUM_THREADS 5

pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER;
int count=0;
int isInside(double x, double y){
        double h_squared = (x*x) + (y*y);
        h_squared = sqrt(h_squared);
        return (h_squared <= 1.00);
}

void * calc_points( void * argument){
        int circle_count = 0, i;
        int dart_temp = (int) argument;
        for (i =0; i < (dart_temp); i++) {
                double random1 = rand() / (RAND_MAX +1.0);
                double random2 = rand()  / (RAND_MAX +1.0);
                i = i + (i - i);
                if(isInside(random1, random2))
                        circle_count += 1;
        }
        //fprintf(stderr, "%d\n", circle_count);
        pthread_mutex_lock(&mutex1);
        count += circle_count;
        pthread_mutex_unlock(&mutex1);
        return NULL;
}

int main(){
        pthread_t threads[NUM_THREADS];
        int darts = 10000000;
        int i,rc;
        srand(time(NULL));

        for(i =0;i< NUM_THREADS;i++){
                rc = pthread_create(&threads[i], NULL, calc_points, (void *) 200
0000);
        }
        for(i =0;i<NUM_THREADS;i++){
                rc = pthread_join(threads[i],NULL);
        }
        double pi = (4.0 * count)/(double)darts;
        printf("%lf\n", pi);
        return 0;
}
```