Code

```
#include <stdio.h>
#include <math.h>
#include <time.h>

int main(int argc, char *argv[]){
    /* Variable declarations */
    clock_t t;
    double time,timesquareroot; // in seconds
    int n;
    double xMid, xDelta, pi= 0, pisqrt = 0,  area = 0;

    /* User input query */
    printf("Enter number of rectangles: ");
    scanf("%d", &n);                        //Enter number of rectangles

    /* Width and Mid point calculations */
    xDelta = (double)1/n;                   //Calculate the the width of the rectangles
    xMid = xDelta/2;                        //Mid point of the rectangles

    t = clock();
    for(double i = 0; i < n; i++){          //loop to sum the area
        area = (4 / (1 + pow(xMid+(xDelta*i),2)));
        pi += xDelta*area;
    }
    t = clock() - t;
    time = ((double)t)/CLOCKS_PER_SEC;

    t = clock();
    for(double i = 0; i < n; i++){          //loop to sum the area using square root
        area = 4 *sqrt(1 - pow(xMid+(xDelta*i),2));
        pisqrt += xDelta*area;
    }
    t = clock() - t;
    timesquareroot = ((double)t)/CLOCKS_PER_SEC;

    printf("PI: %0.12f. Time to execute loop: %lf.\n",pi, time);
    printf("PI Square root: %0.12f. Time to execute loop: %lf.\n",pisqrt, timesquareroot);

    return 0;
}
```

Result

N	Pi	Pi Square root
2	3.162352941176	3.259367328636
20	3.141800986893	3.145430588679
200	3.141594736923	3.141714389345
2000	3.141592674423	3.141592674423
20000	3.141592653798	3.141592775366
200000	3.141592653592	3.141592657441

Discussion

What did you learn?
How to compute pi using integrals.

How does accuracy depend on n?
Having more rectangles will have more accurate result since more it will cover more area than a smaller set of rectangles

How does runtime depend on n?
If there are more rectangles then there will be more calculations to be done thus increases time due to having to do more calculations.

How might you divide the work up if the code is to run on multiple processors concurrently? What coordination or communication would be required between the processors?
Splitting the work of the calculations by dividing the number of areas to the number of processors.Functional parallelism would be required since the data are not dependent on previous value in my code.