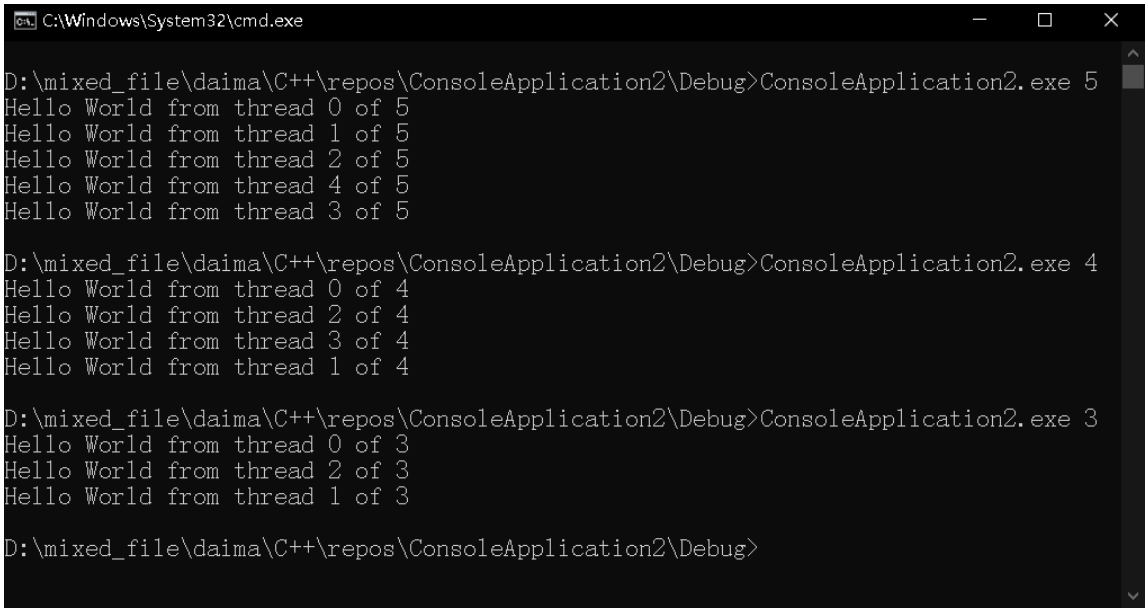
作业 3

第一题（20%）

请编译，并运行以下使用OpenMP 的Hello World 程序三次，逐次打印运行结果。

```
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

void Hello(void); /* Thread function */

int main(int argc, char* argv[]) {
    /* Get number of threads from command line */
    int thread_count = strtol(argv[1], NULL, 10);

# pragma omp parallel num_threads(thread_count)
    Hello();

    return 0;
} /* main *
void Hello(void) {
    int my_rank = omp_get_thread_num();
    int thread_count = omp_get_num_threads();

    printf("Hello World from thread %d of %d\n", my_rank, thread_count);

} /* Hello */
```

答：



第二题（30%）

利用以下公式，使用积分法求π的近似值：

$$\frac{\pi}{4} = \tan^{-1}(-1) = \int_0^1 \frac{dx}{1+x^2}$$

现在分别基于以下两段程序进行针对上式的相关计算。

程序 1：

```
#include <stdio.h> #include
<stdlib.h> #include <sys/time.h>
#include <omp.h>

#define NUM_STEPS 1000000000

int main(int argc, char* argv[]) {
        double x, pi, step, sum=0.0, start, stop; int i;
        step = 1.0/NUM_STEPS;
        struct timeval tv;
        gettimeofday(&tv, NULL); //Record time in milliseconds
        start = (tv.tv_sec)*1000 + (tv.tv_usec)/1000;

        for(i=0; i<NUM_STEPS; i++){
                        x = (i+0.5)*step;
                        sum = sum + 1.0/(1.0+x*x);
        }
                pi = 4.0*sum*step; gettimeofday(&tv,
                NULL);
                stop = (tv.tv_sec)*1000 + (tv.tv_usec)/1000;
                printf("The value of PI is %f, and the total calculation time is %f ms", pi, stop-
 start);
}
```

程序 2：

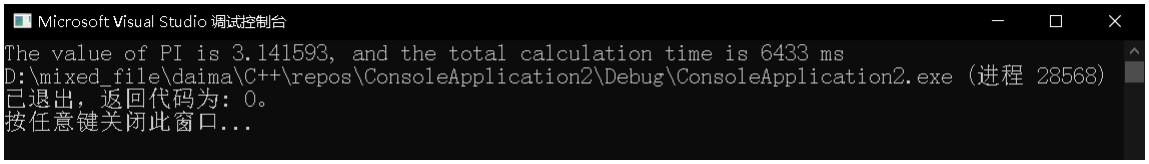
```
#include <stdio.h>
 #include <stdlib.h>
#include <sys/time.h>
 #include <omp.h>

#define NUM_STEPS 1000000000
```

```
int main(int argc, char* argv[]) {
        double x, pi, step, sum=0.0, start, stop; int i;
        step = 1.0/NUM_STEPS;
        struct timeval tv;
        gettimeofday(&tv, NULL); //Record time in milliseconds
        start = (tv.tv_sec)*1000 + (tv.tv_usec)/1000;

#pragma omp parallel for reduction(+:sum) private(x)
        for(i=0; i<NUM_STEPS;i++){
                x = (i+0.5)*step;
                sum = sum + 1.0/(1.0+x*x);
          }
          pi = 4.0*sum*step;

          gettimeofday(&tv, NULL);
          stop = (tv.tv_sec)*1000 + (tv.tv_usec)/1000;
          printf("The value of PI is %f, and the total calculation time is %f ms", pi, stop-start);
}
```

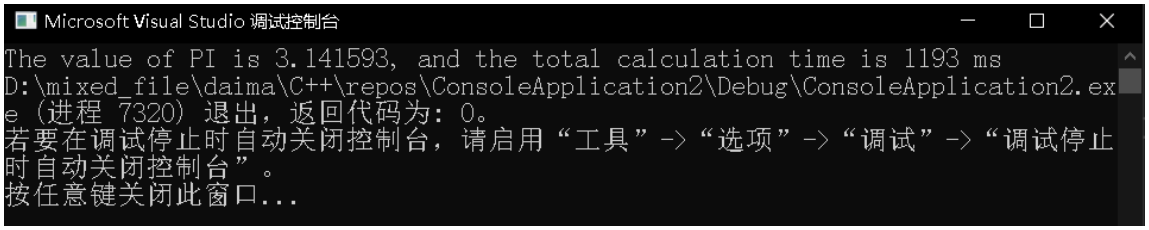
请分别打印基于以上两段程序的计算结果，并统计两段程序执行的时间。

答：

程序一：



程序二：



显然程序二执行比程序一快六倍左右。

第三题（50%）

除了第二题中的方法，我们可以使用蒙特卡洛（Monte Carlo）方法计算π的值。

请参考图 2，其中，阴影部分（1/4 单位圆）的面积为$\frac{\pi}{4}$。我们可以在图 2 中面积为 1 的正方形里（坐标轴和虚线围成的部分）随机生成点，随机点落在阴影部分的概率为$\frac{\pi}{4}$。若点的数量足够，我们可以经由统计模拟方法计算得到π的值。

基于以上分析，写一个基于 OpenMP 的并行程序，完成该π值的计算任务，并报告你的实验结果（表 1 供参考，你也可以采用自己的呈现方式）。

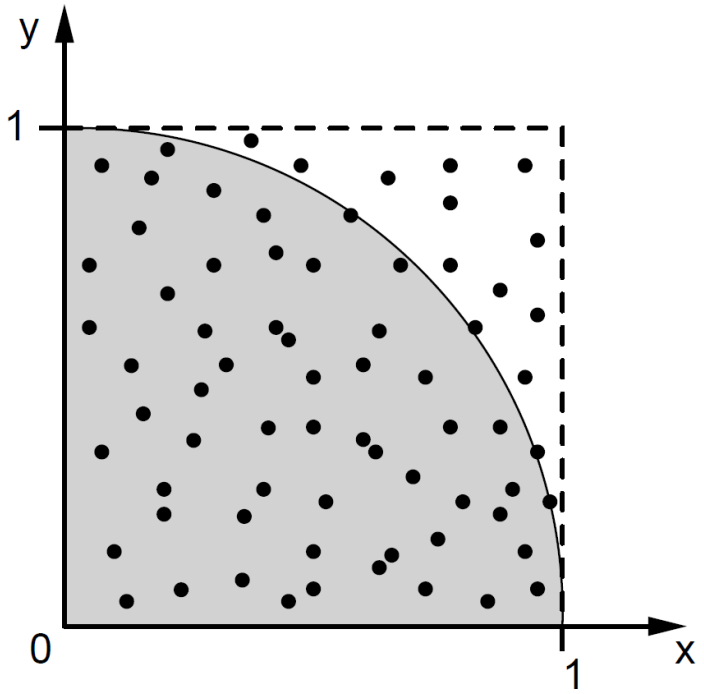
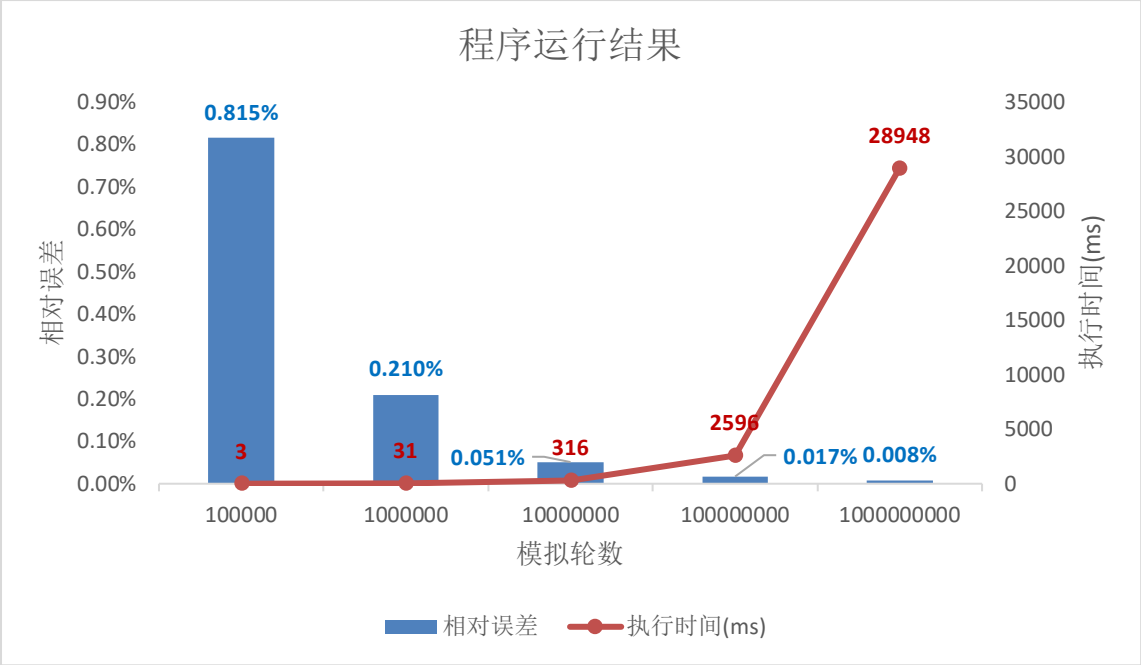


图 2

答：

随机点数目 （模拟轮数）	相对误差 $\frac{\lvert x-x^* \rvert}{x^*}$ （x、x^*为估计值和真实值）	程序运行时间
100000	0.815%	3
1000000	0.210%	31
10000000	0.051%	316
100000000	0.017%	2596
1000000000	0.008%	28948

表 1



图表 1运行截图

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.18363.1556]
(c) 2019 Microsoft Corporation。保留所有权利。

D:\mixed_file\daima\C++\repos\ConsoleApplication2\Debug>ConsoleApplication2.exe 100000

The value of PI is 3.113760
the relative error is 0.886%
the step is 100000
the total calculation time is 3 ms

D:\mixed_file\daima\C++\repos\ConsoleApplication2\Debug>ConsoleApplication2.exe 1000000

The value of PI is 3.136788
the relative error is 0.153%
the step is 1000000
the total calculation time is 31 ms

D:\mixed_file\daima\C++\repos\ConsoleApplication2\Debug>ConsoleApplication2.exe 10000000

The value of PI is 3.139864
the relative error is 0.055%
the step is 10000000
the total calculation time is 283 ms

D:\mixed_file\daima\C++\repos\ConsoleApplication2\Debug>ConsoleApplication2.exe 100000000

The value of PI is 3.141061
the relative error is 0.017%
the step is 100000000
the total calculation time is 2605 ms

D:\mixed_file\daima\C++\repos\ConsoleApplication2\Debug>ConsoleApplication2.exe 1000000000

The value of PI is 3.141352
the relative error is 0.008%
the step is 1000000000
the total calculation time is 28771 ms

D:\mixed_file\daima\C++\repos\ConsoleApplication2\Debug>
```

相关代码如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/timeb.h>
#include <omp.h>
#include<time.h>
#include<math.h>

#define PI 3.14159265
int main(int argc, char* argv[]) {
        int num = strtol(argv[1], NULL, 10);
        double x,y,pi; int i,k; long t,sum=0;
        timeb t1, t2;
        ftime(&t1);
        srand((int)time(0));

        # pragma omp parallel for reduction(+:sum) private(x,y,k)
                        for(i=0; i<num; i++){
                                x = (double)rand() / RAND_MAX;
                                y = (double)rand() / RAND_MAX;
                                if (x*x + y * y <= 1) sum ++;
                        }

        ftime(&t2);
        t = (t2.time - t1.time) * 1000 + (t2.millitm - t1.millitm);
        pi = sum / (double)num * 4;
        printf("\nThe value of PI is %f\nthe relative error is %.3f%%\nthe step is %d\nthe
total calculation time is %ld ms\n", pi,fabs(PI-pi)/PI*100 ,num, t);
}
```