4. Let's revisit computing the value of pi, but this time we will use a series. For instance, we provide you code for the Leibniz's series, developed by Jose Cintra. Implement this series on the GPU, allowing the user to enter the number of iterations. Make sure to compute find an efficient computation of this kernel that utilizes the parallelism provided on the GPU. Then modify this code to use single precision math. Show results for at least 10 different number of iterations of the series and discuss how precision plays a role in the rate of convergence.

1) Test number: 640

```
Test number:640
Results: 3.1062500000000002
```

2) Test number: 12800

```
Test number:12800
Results: 3.1549999999999998
```

3) Test number: 25600

```
Test number:256000
Results: 3.1448593749999998
```

4) Test number: 256000

```
Test number:2560000
Results: 3.1414296875000001
```

5) Test number: 5120000

```
Test number:5120000
Results: 3.1420109374999998
```

6) Test number: 51200000

```
Test number:51200000
Results: 3.1419493749999998
```

7) Test number: 2560000000

```
Test number:2560000000
Results: 3.1415388000000002
```

8) Test number: 5120000000

```
Test number:5120000000
Results: 3.1415769999999998
```

9) Test number: 10240000000

```
Test number:10240000000
Results: 3.1415928000000002
```

10) Test number: 1024000000000

```
Test number:1024000000000
Results: 3.1415920640000001
```

It can be seen that as the number of iteration increases, the estimated pi value becomes more and more accurate.