1. Compute Pi

Method:

The calculation is based on the idea of indefinite integral.

Integral 1 / (1 + X^2) dx = Arctan (X)

Hence, the definite integral

Integral (0 <= X <= 1) 1 / (1 + X^2) dx = Arctan (1) - Arctan (0)
= PI / 4.

A standard way to approximate an integral uses the midpoint rule.

Midpoint Rule

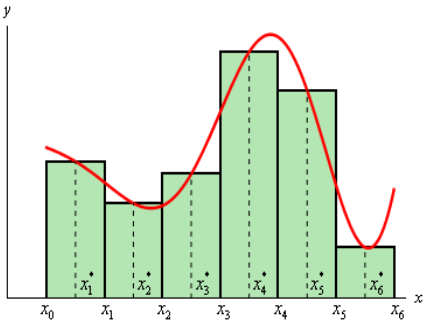
This is the rule that should be somewhat familiar to you. We will divide the interval $[a,b]$ into n subintervals of equal width,

$$\Delta x = \frac{b-a}{n}$$

We will denote each of the intervals as follows,

$$[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n] \quad \text{where } x_0 = a \text{ and } x_n = b$$

Then for each interval let x_i^* be the midpoint of the interval. We then sketch in rectangles for each subinterval with a height of $f(x_i^*)$. Here is a graph showing the set up using $n = 6$.



We can easily find the area for each of these rectangles and so for a general n we get that,

$$\int_a^b f(x)\,dx \approx \Delta x f(x_1^*) + \Delta x f(x_2^*) + \cdots + \Delta x f(x_n^*)$$

Or, upon factoring out a Δx we get the general Midpoint Rule.

$$\boxed{\int_a^b f(x)\,dx \approx \Delta x \left[f(x_1^*) + f(x_2^*) + \cdots + f(x_n^*) \right]}$$

If we create N equally spaced intervals of width 1/N, then the midpoint of the I-th interval is

X(I) = (2*I-1)/(2*N).

The approximation for the integral is then:

Sum (1 <= I <= N) (1/N) * 1 / (1 + X(I)^2)

In order to compute PI, we multiply this by 4; we also can pull out the factor of 1/N, so that the formula you see in the program looks like:

(4 / N) * Sum (1 <= I <= N) 1 / (1 + X(I)^2)

Sequential Version:

```
sum = 0
for ( i = 1; i <= n; i++ )
{
   x = ( 2 * i - 1 )/ (2*n)
   sum = sum + 1 / ( 1 + x * x );
}

 estimated_pi = 4 * sum /  n ;
```

Repeat the algorithm for different **n** values ranging from 1-1^9.

Output:

Error difference with different n values

N	Mode	Estimate	Error	Time
1	SEQ	3.2	0.0584073	5.64003e-07
10	SEQ	3.14243	0.000833331	2.9098e-07
100	SEQ	3.1416	8.33333e-06	1.95298e-06
1000	SEQ	3.14159	8.33333e-08	1.8593e-05
10000	SEQ	3.14159	8.33341e-10	0.000193436
100000	SEQ	3.14159	8.36842e-12	0.00187383
1000000	SEQ	3.14159	2.84217e-14	0.0185919
10000000	SEQ	3.14159	6.21725e-14	0.186129
100000000	SEQ	3.14159	6.33271e-13	1.56283
1000000000	SEQ	3.14159	1.77636e-13	14.648

OpenMP Version:

```
# pragma omp parallel shared ( n ) private ( i, x )

# pragma omp for reduction ( + : sum )

 for ( i = 1; i <= n; i++ )
 {
        x = ( 2 * i - 1 )/ (2*n)
           sum = sum + 1 / ( 1 + x * x );
 }

 estimated_pi = 4 * sum /  n ;
```

MPI Version:

```
//Broadcast all the processor the value of number of iterations needs to be performed.
MPI_Bcast ( &n, 1, MPI_INT, 0, MPI_COMM_WORLD );

//estimate pi
estimate_pi = r8_pi_est( n, id, p );

//find the average
MPI_Reduce ( &estimate, &estimate_pi, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD );
estimate = estimate/p;
```