DEFINING THE INTEGRAND f[x], THE LIMITS OF INTEGRATION a AND b, AND COMPUTING THE EXACT VALUE OF THE INTEGRAL

```
In[570]:= f[x_] = Sqrt[x]
```

Out[570]= $\sqrt{x}$

```
In[571]:= a = 4
Out[571]= 4

In[572]:= b = 9
Out[572]= 9

In[573]:= exactValue = Integrate[f[x], {x, a, b}]
```

Out[573]= $\frac{38}{3}$

LEFT RIEMANN SUM

```
In[574]:= numPoints = 10;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          leftRiemannSum = Sum[f[x[i]] * DeltaX, {i, 0, numPoints - 1}];
          N[leftRiemannSum - exactValue, 8]

Out[576]= -0.25173523

In[577]:= numPoints = 100;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          leftRiemannSum = Sum[f[x[i]], {i, 0, numPoints - 1}] * DeltaX;
          N[leftRiemannSum - exactValue, 8]

Out[579]= -0.025017361

In[516]:= numPoints = 1000;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ ; x[i] = a + i * DeltaX];
          leftRiemannSum = Sum[f[x[i]], {i, 0, numPoints - 1}] * DeltaX;
          N[leftRiemannSum - exactValue, 8]

Out[518]= -0.0025001736

In[580]:= numPoints = 10 000;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ ; x[i] = a + i * DeltaX];
          leftRiemannSum = Sum[f[x[i]], {i, 0, numPoints - 1}] * DeltaX;
          N[leftRiemannSum - exactValue, 8]

Out[582]= -0.00025000174
```

In[583]:=
```
numPoints = 100 000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ ; x[i] = a + i * DeltaX];
leftRiemannSum = Sum[f[x[i]], {i, 0, numPoints - 1}] * DeltaX;
N[leftRiemannSum - exactValue, 8]
```
Out[585]= -0.000025000017

RIGHT RIEMANN SUM

In[586]:=
```
numPoints = 10;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
leftRiemannSum = Sum[f[x[i + 1]], {i, 0, numPoints - 1}] * DeltaX;
N[leftRiemannSum - exactValue, 8]
```
Out[588]= 0.24826477

In[589]:=
```
numPoints = 100;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
leftRiemannSum = Sum[f[x[i + 1]], {i, 0, numPoints - 1}] * DeltaX;
N[leftRiemannSum - exactValue, 8]
```
Out[591]= 0.024982639

In[592]:=
```
numPoints = 1000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
leftRiemannSum = Sum[f[x[i + 1]], {i, 0, numPoints - 1}] * DeltaX;
N[leftRiemannSum - exactValue, 8]
```
Out[594]= 0.0024998264

In[595]:=
```
numPoints = 10 000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
leftRiemannSum = Sum[f[x[i + 1]], {i, 0, numPoints - 1}] * DeltaX;
N[leftRiemannSum - exactValue, 8]
```
Out[597]= 0.00024999826

In[598]:=
```
numPoints = 100 000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
leftRiemannSum = Sum[f[x[i + 1]], {i, 0, numPoints - 1}] * DeltaX;
N[leftRiemannSum - exactValue, 8]
```
Out[600]= 0.000024999983

MIDPOINT RULE

```
In[601]:= numPoints = 10;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          midpointRule = Sum[f[(x[i] + x[i + 1]) / 2] * DeltaX, {i, 0, numPoints - 1}];
          N[midpointRule - exactValue, 8]

Out[603]= 0.00086728577

In[604]:= numPoints = 100;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          midpointRule = Sum[f[(x[i] + x[i + 1]) / 2] * DeltaX, {i, 0, numPoints - 1}];
          N[midpointRule - exactValue, 8]
```

Out[606]= 8.6804783×10^{-6}

```
In[607]:= numPoints = 1000;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          midpointRule = Sum[f[(x[i] + x[i + 1]) / 2] * DeltaX, {i, 0, numPoints - 1}];
          N[midpointRule - exactValue, 8]
```

Out[609]= 8.6805548×10^{-8}

```
In[610]:= numPoints = 10 000;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          midpointRule = Sum[f[(x[i] + x[i + 1]) / 2] * DeltaX, {i, 0, numPoints - 1}];
          N[midpointRule - exactValue, 8]
```

Out[612]= $8.6805555 \times 10^{-10}$

```
In[613]:= numPoints = 100 000;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
          midpointRule = Sum[f[(x[i] + x[i + 1]) / 2] * DeltaX, {i, 0, numPoints - 1}];
          N[midpointRule - exactValue, 8]
```

Out[615]= $8.6805556 \times 10^{-12}$

TRAPEZOIDAL RULE

```
In[616]:= numPoints = 10;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];  midpointRule =
           (f[x[0]] + 2 * Sum[f[x[i]], {i, 1, numPoints - 1}] + f[x[numPoints]]) * DeltaX / 2;
          N[midpointRule - exactValue, 8]

Out[618]= -0.0017352310

In[619]:= numPoints = 100;  DeltaX = (b - a) / numPoints;
          Array[x, numPoints + 1, 0];
          For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];  midpointRule =
           (f[x[0]] + 2 * Sum[f[x[i]], {i, 1, numPoints - 1}] + f[x[numPoints]]) * DeltaX / 2;
          N[midpointRule - exactValue, 8]

Out[621]= -0.000017361023
```

In[622]:=
```
numPoints = 1000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];  midpointRule =
 (f[x[0]] + 2 * Sum[f[x[i]], {i, 1, numPoints - 1}] + f[x[numPoints]]) * DeltaX / 2;
N[midpointRule - exactValue, 8]
```

Out[624]= $-1.7361110 \times 10^{-7}$

In[625]:=
```
numPoints = 10 000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];  midpointRule =
 (f[x[0]] + 2 * Sum[f[x[i]], {i, 1, numPoints - 1}] + f[x[numPoints]]) * DeltaX / 2;
N[midpointRule - exactValue, 8]
```

Out[627]= $-1.7361111 \times 10^{-9}$

In[628]:=
```
numPoints = 100 000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];  midpointRule =
 (f[x[0]] + 2 * Sum[f[x[i]], {i, 1, numPoints - 1}] + f[x[numPoints]]) * DeltaX / 2;
N[midpointRule - exactValue, 8]
```

Out[630]= $-1.7361111 \times 10^{-11}$

SIMPSON ' S RULE

In[631]:=
```
numPoints = 10;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
simpsonRule = (f[x[0]] + 2 * Sum[f[x[2 * i]], {i, 1, numPoints / 2 - 1}] +
     4 * Sum[f[x[2 * i + 1]], {i, 0, numPoints / 2 - 1}] + f[x[numPoints]]) *
   DeltaX / 3;  N[simpsonRule - exactValue, 8]
```

Out[633]= $-3.4711579 \times 10^{-6}$

In[634]:=
```
numPoints = 100;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
simpsonRule = (f[x[0]] + 2 * Sum[f[x[2 * i]], {i, 1, numPoints / 2 - 1}] +
     4 * Sum[f[x[2 * i + 1]], {i, 0, numPoints / 2 - 1}] + f[x[numPoints]]) *
   DeltaX / 3;  N[simpsonRule - exactValue, 8]
```

Out[636]= $-3.5325289 \times 10^{-10}$

In[637]:=
```
numPoints = 1000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
simpsonRule = (f[x[0]] + 2 * Sum[f[x[2 * i]], {i, 1, numPoints / 2 - 1}] +
     4 * Sum[f[x[2 * i + 1]], {i, 0, numPoints / 2 - 1}] + f[x[numPoints]]) *
   DeltaX / 3;  N[simpsonRule - exactValue, 8]
```

Out[639]= $-3.5331672 \times 10^{-14}$

In[640]:=
```
numPoints = 10000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
simpsonRule = (f[x[0]] + 2 * Sum[f[x[2 * i]], {i, 1, numPoints / 2 - 1}] +
     4 * Sum[f[x[2 * i + 1]], {i, 0, numPoints / 2 - 1}] + f[x[numPoints]]) *
   DeltaX / 3;  N[simpsonRule - exactValue, 8]
```

Out[642]= $-3.5331736 \times 10^{-18}$

In[643]:=
```
numPoints = 100000;  DeltaX = (b - a) / numPoints;
Array[x, numPoints + 1, 0];
For[ i = 0 , i ≤ numPoints, i++ , x[i] = a + i * DeltaX];
simpsonRule = (f[x[0]] + 2 * Sum[f[x[2 * i]], {i, 1, numPoints / 2 - 1}] +
     4 * Sum[f[x[2 * i + 1]], {i, 0, numPoints / 2 - 1}] + f[x[numPoints]]) *
   DeltaX / 3;  N[simpsonRule - exactValue, 8]
```

Out[645]= $-3.5331737 \times 10^{-22}$