

APP WEEK-9 LAB

Q1. Calculate root 2 with 100 decimals

Code:

```
import sympy

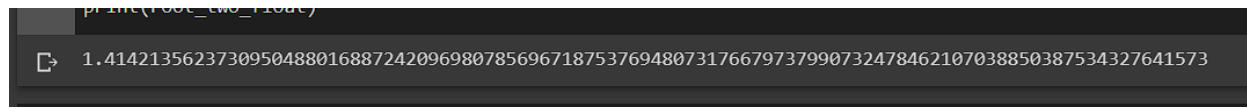
# Set the precision to 100 decimal places
sympy_float = lambda x: sympy.Float(x, 100)

# Calculate the square root of 2
root_two = sympy.sqrt(2)

# Evaluate the square root of 2 to 100 decimal places
root_two_float = root_two.evalf(100)

# Print the result
print(root_two_float)
```

SnapShot:



```
print(root_two_float)
1.414213562373095048801688724209698078569671875376948073176679737990732478462107038850387534327641573
```

Q2. Calculate $1/2 + 1/3$ in rational arithmetic

Code:

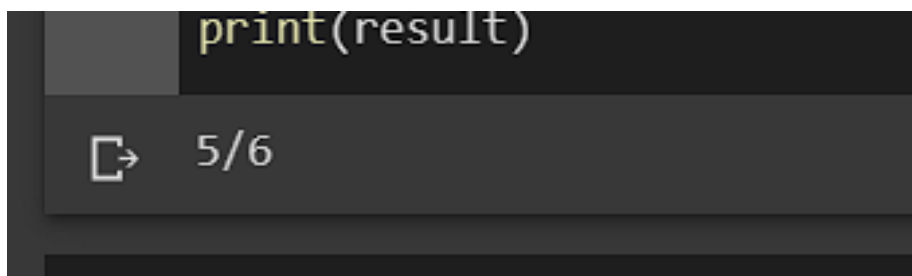
```
from fractions import Fraction

# Define the fractions
frac1 = Fraction(1, 2)
frac2 = Fraction(1, 3)

# Add the fractions
result = frac1 + frac2

# Print the result
print(result)
```

SnapShot:



```
print(result)
5/6
```

Q3. Calculate the expanded form of $(x+y)^6$

Code:

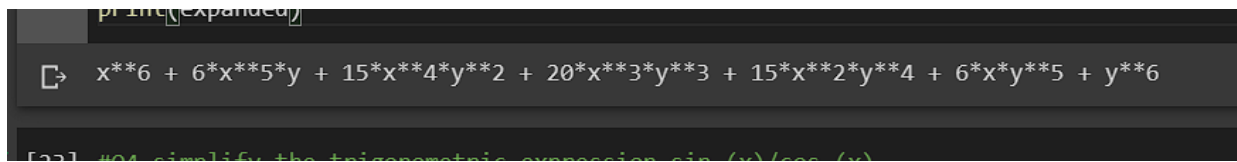
```
import sympy

# Define variables
x, y = sympy.symbols('x y')

# Expand the expression
expanded = sympy.expand((x + y)**6)

# Print the result
print(expanded)
```

SnapShot:

A screenshot of a Jupyter Notebook cell. The top part shows a code editor with the text 'print(expanded)'. Below it is a console output area showing the expanded polynomial: $x^6 + 6x^5y + 15x^4y^2 + 20x^3y^3 + 15x^2y^4 + 6xy^5 + y^6$. At the bottom, a green prompt '[22]:' is visible, followed by a comment in green text: '#04 simplify the trigonometric expression sin (x)/cos (x)'.

Q4. Simplify the trigonometric expression $\sin(x)/\cos(x)$

Code:

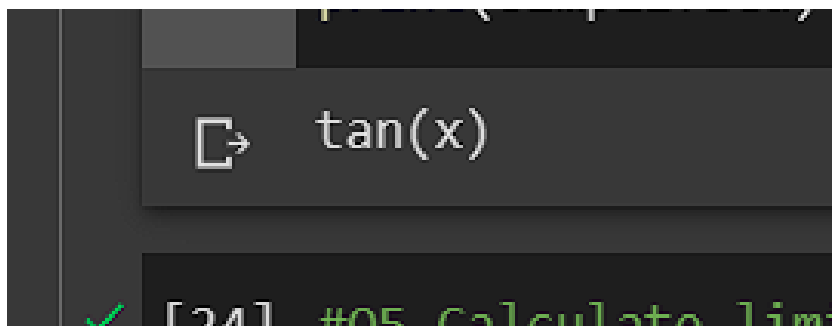
```
import sympy

# Define variable
x = sympy.symbols('x')

# Simplify the expression
simplified = sympy.tan(x)

# Print the result
print(simplified)
```

SnapShot:

A screenshot of a Jupyter Notebook cell. The top part shows a code editor with the text 'tan(x)'. Below it is a console output area showing the simplified expression: $\tan(x)$. At the bottom, a green prompt '[24]:' is visible, followed by a comment in green text: '#05 Calculate limit'.

Q5. Calculate limit x tends to 0 $((\sin(x)-x)/x^3)$

Code:

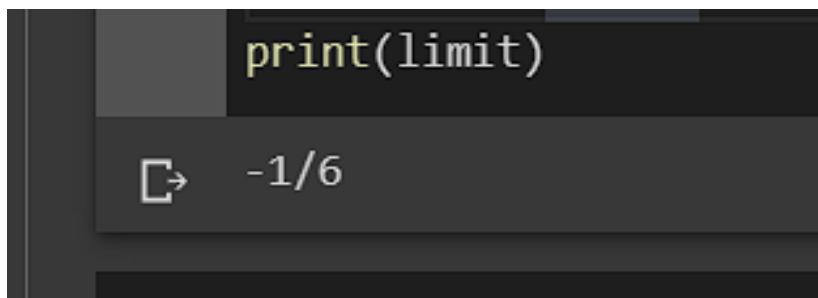
```
import sympy

# Define the variable and expression
x = sympy.symbols('x')
expr = (sympy.sin(x) - x) / x**3

# Calculate the limit as x approaches 0
limit = sympy.limit(expr, x, 0)

# Print the result
print(limit)
```

SnapShot:



Q6. Calculate the derivative of $\log(x)$, $1/x$, $\sin(x)$, $\cos(x)$ for x

Code:

```
import sympy

# Define variable
x = sympy.symbols('x')

# Calculate the derivatives
derivative_log = sympy.diff(sympy.log(x), x)
derivative_inv = sympy.diff(1/x, x)
derivative_sin = sympy.diff(sympy.sin(x), x)
derivative_cos = sympy.diff(sympy.cos(x), x)

# Print the results
print("The derivative of log(x) is:", derivative_log)
print("The derivative of 1/x is:", derivative_inv)
print("The derivative of sin(x) is:", derivative_sin)
print("The derivative of cos(x) is:", derivative_cos)
```

SnapShot:

```
print( "The derivative of sin(x) is: ", derivative_s
print("The derivative of cos(x) is:", derivative_c

☐ The derivative of log(x) is: 1/x
The derivative of 1/x is: -1/x**2
The derivative of sin(x) is: cos(x)
The derivative of cos(x) is: -sin(x)
```

Q7. Solve the system of equations $x+y=2$, $2x+y=0$

Code:

```
import sympy

# Define the variables
x, y = sympy.symbols('x y')

# Define the equations
eq1 = sympy.Eq(x + y, 2)
eq2 = sympy.Eq(2*x + y, 0)

# Solve the system of equations
sol = sympy.solve((eq1, eq2), (x, y))

# Print the solution
print("The solution is:", sol)
```

SnapShot:

```
print("The solution is:", sol)

☐ The solution is: {x: -2, y: 4}
```

✓ [27] #Q8 Integrate x^2 , $\sin(x)$, $\cos(x)$ i

Q8. Integrate x^2 , $\sin(x)$, $\cos(x)$ in terms of x and y

Code:

```
import sympy

# Define variables
x, y = sympy.symbols('x y')

# Integrate  $x^2$  with respect to  $x$ 
integrate_x2_x = sympy.integrate(x**2, x)
print("The integral of  $x^2$  with respect to  $x$  is:", integrate_x2_x)

# Integrate  $x^2$  with respect to  $y$ 
integrate_x2_y = sympy.integrate(x**2, y)
print("The integral of  $x^2$  with respect to  $y$  is:", integrate_x2_y)

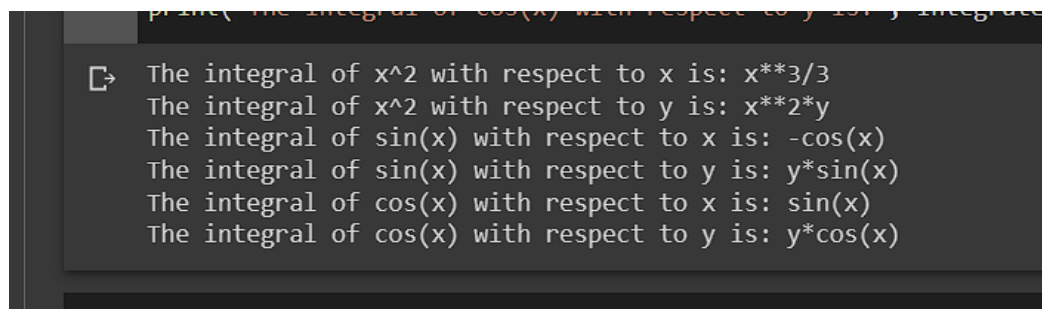
# Integrate  $\sin(x)$  with respect to  $x$ 
integrate_sin_x = sympy.integrate(sympy.sin(x), x)
print("The integral of  $\sin(x)$  with respect to  $x$  is:", integrate_sin_x)

# Integrate  $\sin(x)$  with respect to  $y$ 
integrate_sin_y = sympy.integrate(sympy.sin(x), y)
print("The integral of  $\sin(x)$  with respect to  $y$  is:", integrate_sin_y)

# Integrate  $\cos(x)$  with respect to  $x$ 
integrate_cos_x = sympy.integrate(sympy.cos(x), x)
print("The integral of  $\cos(x)$  with respect to  $x$  is:", integrate_cos_x)

# Integrate  $\cos(x)$  with respect to  $y$ 
integrate_cos_y = sympy.integrate(sympy.cos(x), y)
print("The integral of  $\cos(x)$  with respect to  $y$  is:", integrate_cos_y)
```

SnapShot:



Q9. Solve $f'(x) + 9f(x)=1$

Code:

```
import sympy

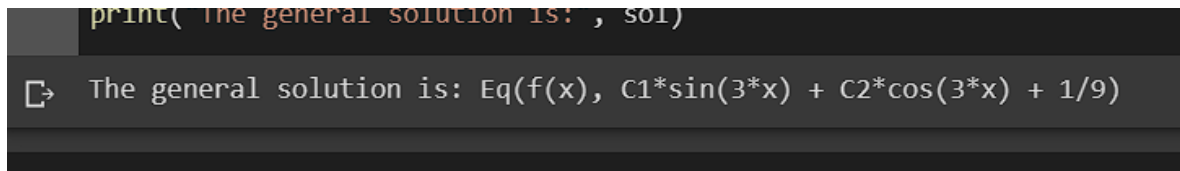
# Define the function and the variable
f = sympy.Function('f')
x = sympy.Symbol('x')

# Define the differential equation
deq = sympy.diff(f(x), x, 2) + 9*f(x) - 1

# Solve the differential equation
sol = sympy.dsolve(deq, f(x))

# Print the solution
print("The general solution is:", sol)
```

SnapShot:



```
print("The general solution is:", sol)

The general solution is: Eq(f(x), C1*sin(3*x) + C2*cos(3*x) + 1/9)
```

Q10. Using matrices solve the linear equations: $3x+7y=12z$ and $4x-2y=5z$

Code:

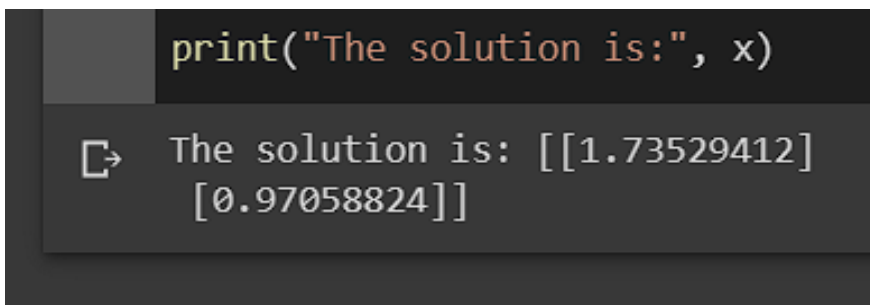
```
import numpy as np

# Define the coefficient matrix and the right-hand side vector
A = np.array([[3, 7], [4, -2]])
b = np.array([[12], [5]])

# Solve the system of equations using matrix inversion
x = np.linalg.inv(A) @ b

# Print the solution
print("The solution is:", x)
```

SnapShot:



```
print("The solution is:", x)

The solution is: [[1.73529412]
 [0.97058824]]
```