**Ex.No:11**

**Date:**

**IMPLEMENTATION OF SYMBOLIC PROGRAMMING PARADIGM**

**AIM:**

To implement symbolic programming paradigm in python.

13a. Write the commands to perform the operations on substitutions and expressions

**ALGORITHM:**

1. Import sympy module
2. Evaluate the expression using sympy command
3. Print the result

**Program:**

# import sympy

from sympy import \*

x, y, z = symbols('x y z')

exp = x\*\*2 + 7 \* y + z

print("Before Substitution : {}".format(exp))

# Use sympy.subs() method

res\_exp = exp.subs([(x, 2), (y, 4), (z, 1)])

print("After Substitution : {}".format(res\_exp))

x = symbols('x')

exp = cos(x) + 7

print("Before Substitution : {}".format(exp))

# Use sympy.subs() method

res\_exp = exp.subs(x, 0)

print("After Substitution : {}".format(res\_exp))

x, y = symbols('x y')

exp = x\*\*2 + 1

print("Before Substitution : {}".format(exp))

# Use sympy.subs() method

res\_exp = exp.subs(x, y)

print("After Substitution : {}".format(res\_exp))

**Output:**

Before Substitution : x\*\*2 + 7\*y + z

After Substitution : 33

Before Substitution : cos(x) + 7

After Substitution : 8

Before Substitution : x\*\*2 + 1

After Substitution : y\*\*2 + 1

13b. To perform the following operations on matrices

**ALGORITHM:**

1. Import matrix from sympy.matrices.
2. Create the matrix
3. Print the matrix
4. Display the matrix
5. Display 0th row
6. Print first column
7. Delete the first column from the matrix
8. Insert the row into the matrix
9. Generate two matrices
10. Print addition of two matrices
11. Print the multiplication of two matrices

**Program:**

from sympy.matrices import Matrix

m=Matrix([[1,2,3],[2,3,1]])

print(m)

M=Matrix(2,3,[10,40,30,2,6,9])

print(M)

print(M.shape)

print(M.row(0))

M.col(1)

M.row(1)[1:3]

print(M)

M=Matrix(2,3,[10,40,30,2,6,9])

M.col\_del(1)

a=Matrix([[1,2,3],[2,3,1]])

print(a)

a1=Matrix([[10,30]])

a=M.row\_insert(0,M)

print(a)

a2=Matrix([40,6])

a=M.col\_insert(1,M)

print(a)

M1=Matrix([[1,2,3],[3,2,1]])

M2=Matrix([[4,5,6],[6,5,4]])

print(M1+M2)

M1=Matrix([[1,2,3],[3,2,1]])

M2=Matrix([[4,5],[6,6],[5,4]])

print(M1\*M2)

**Output:**

Matrix([[1, 2, 3], [2, 3, 1]])

Matrix([[10, 40, 30], [2, 6, 9]])

(2, 3)

Matrix([[10, 40, 30]])

Matrix([[10, 40, 30], [2, 6, 9]])

Matrix([[1, 2, 3], [2, 3, 1]])

Matrix([[10, 30], [2, 9], [10, 30], [2, 9]])

Matrix([[10, 10, 30, 30], [2, 2, 9, 9]])

Matrix([[5, 7, 9], [9, 7, 5]])

Matrix([[31, 29], [29, 31]])

13c. Write the commands to find derivative, integration, limits, quadratic equation

**ALGORITHM:**

1. Import sympy module
2. Make a symbol
3. Find the derivative of the expression
4. Print the result
5. Find the integration of the expression
6. Print the result
7. Find the limit of the expression
8. Print the result
9. Find the quadratic equation of the expression
10. Print the result

**Program:**

from sympy import \*

x = Symbol('x')

 #make the derivative of cos(x)\*e ^ x

ans1 = diff(cos(x)\*exp(x), x)

print("The derivative of the  sin(x)\*e ^ x : ", ans1)

# Compute (e ^ x \* sin(x)+ e ^ x \* cos(x))dx

ans2 = integrate(exp(x)\*sin(x) + exp(x)\*cos(x), x)

print("The result of  integration is : ", ans2)

# Compute definite integral of sin(x ^ 2)dx

# in b / w interval of ? and ?? .

ans3 = integrate(sin(x\*\*2), (x, -oo, oo))

print("The value of integration is : ", ans3)

# Find the limit of sin(x) / x given x tends to 0

ans4 = limit(sin(x)/x, x, 0)

print("limit is : ", ans4)

# Solve quadratic equation like, example : x ^ 2?2 = 0

ans5 = solve(x\*\*2 - 2, x)

print("roots are : ", ans5)

**Output:**

The derivative of the sin(x)\*e ^ x : -exp(x)\*sin(x) + exp(x)\*cos(x)

The result of integration is : exp(x)\*sin(x)

The value of integration is : sqrt(2)\*sqrt(pi)/2

* limit is : 1

roots are : [-sqrt(2), sqrt(2)]

RESULT:

Thus the Python program to implement symbolic program have been written and executed successfully.