

**Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования «Московский
технический университет связи и информатики» (МТУСИ)**

ЛАБОРАТОРНАЯ РАБОТА ПО ТЕМЕ

**«Символьные арифметические выражений
и их элементы»**

Выполнил:

Петренко Данила (БИБ2206)

Номер выполняемой работы: 3.4

Вариант:21

Проверил:

2023г.

3.4.1 Общее задание

- 1) Изучить учебные материалы по Теме 3.3 и 3.4.
- 2) Выбрать вариант индивидуального задания из таблиц. 3.1, 3.2, 3.3.
- 3) Создать символьные переменные и определить для них необходимые свойства (предположения).
- 4) Для всех заданий своего варианта создать символьные выражения (всего пять) и получить их символьное математическое
- 5) Оценить символьные арифметические выражений своего варианта и доступ к их атомарным элементам.
- 6) Для каждого символьного выражения оценить численные значения выражения для нескольких значений независимых символов (для каждого выражения не менее 3-х) различными способами.
- 7) Предоставить результаты работы преподавателю и ответить на поставленные вопросы.
- 8) Оформить отчет по выполненной работе.

3.4.2 Индивидуальное задание.

$$b = \frac{x^y}{1 - \frac{1}{e^{-x+\sin y}}}$$

$$t = \cos \frac{\frac{\pi}{7} * \sin(x - 8y)^2}{2.7(x - \pi)}$$

$$f = -5x^2 + y^3 + 2x^3y - x^3\sqrt[3]{3x - y}$$

$$f_1(x) = 2x^2 - e^{\frac{x}{2}}$$

$$f_2(x) = 2 * \arctg(x) - 3x + 2$$

3.4.2 Ход работы.

```
from sympy import *  
import numpy as np  
x = Symbol("x", positive=True, integer=True, nonzero=True)  
y = Symbol("y", float=True, integer=True)
```

✓ 0.2s

```
b = (x**y)/(1-((1)/(exp(-x+sin(y)))))  
b
```

✓ 0.1s

$$\frac{x^y}{1 - e^{x - \sin(y)}}$$

```
simplify(b)
```

✓ 0.1s

$$\frac{x^y}{1 - e^{x - \sin(y)}}$$

```
b.atoms()
```

✓ 0.0s

{-1, 1, x, y}

```
b.evalf()
```

✓ 0.0s

$$\frac{x^y}{1.0 - e^{x - \sin(y)}}$$

```
b.subs(x,10).subs(y,10).evalf(n=5)
```

✓ 0.0s

-2.6351 · 10⁵

```
func = lambdify([x,y],b,'numpy')
variables1 = np.arange(0,8);
variables2 = np.arange(5,13);
func(variables1,variables2)
```

1 ✓ 0.0s

```
array([-0.00000000e+00, -3.85425059e-01, -4.52205872e+01, -1.01436146e+03,
       -7.45629743e+03, -3.83410518e+04, -3.31133290e+05, -5.10237653e+05])
```

✓ type(b)

1 ✓ 0.0s

```
sympy.core.mul.Mul
```

```
t = cos(pi/7)*((sin(x-8*y)**2)/(2.7*(x-pi)))
t
```

✓ 0.0s

$$\frac{\sin^2(x - 8y) \cos\left(\frac{\pi}{7}\right)}{2.7x - 2.7\pi}$$

```
simplify(t)
```

✓ 0.8s

$$\frac{0.37037037037037 \sin^2(x - 8y) \cos\left(\frac{\pi}{7}\right)}{x - \pi}$$

```
t.atoms()
```

✓ 0.0s

```
{-1, -2.70000000000000, -8, 1/7, 2, 2.70000000000000, pi, x, y}
```

```
t.evalf()
```

✓ 0.0s

$$\frac{0.900968867902419 \sin^2(x - 8y)}{2.7x - 8.48230016469244}$$

```
t.subs(x,10).subs(y,10).evalf(n=5)
```

✓ 0.0s

0.029139

```
func = lambdify([x,y],t,'numpy')
variables1 = np.arange(0,8);
variables2 = np.arange(5,13);
func(variables1,variables2)
```

✓ 0.0s

```
array([-0.05897129, -0.00237934, -0.09127054, -2.19971006,  0.31342621,
        0.02700098,  0.01145368,  0.06397421])
```

```
type(t)
```

✓ 0.0s

sympy.core.mul.Mul

```
f = -5*x**2+y**3+2*x**3*y-x**3*(abs(2*x-y))**(1/3)
f
```

✓ 0.1s

$2x^3y - x^3|2x - y|^{0.333333333333333} - 5x^2 + y^3$

```
simplify(f)
```

✓ 0.1s

$2x^3y - x^3|2x - y|^{0.333333333333333} - 5x^2 + y^3$

```
f.atoms()
```

✓ 0.1s

{-1, -5, 0.333333333333333, 2, 3, x, y}

```
f.evalf()
```

✓ 0.0s

$2.0x^3y - x^3|2x - y|^{0.333333333333333} - 5.0x^2 + y^3$

```
f.subs(x,10).subs(y,10).evalf(n=5)
```

✓ 0.1s

18346.0

```
func = lambdify([x,y],f,'numpy')
variables1 = np.arange(0,8);
variables2 = np.arange(5,13);
func(variables1,variables2)
```

✓ 0.0s

array([[125. , 221.41259895, 423.46200344, 864.98213165,
 1737. , 3375. , 5687. , 9282.84707989]])

```
type(f)
1 ✓ 0.1s
sympy.core.add.Add
```

```
f1 = 2*x**2-exp(x/2)
f1
✓ 0.1s
 $2x^2 - e^{\frac{x}{2}}$ 
```

```
simplify(f1)
✓ 0.1s
 $2x^2 - e^{\frac{x}{2}}$ 
```

```
f1.atoms()
✓ 0.1s
{-1, 1/2, 2, x}
```

```
f1.evalf()
✓ 0.1s
 $2.0x^2 - e^{\frac{x}{2}}$ 
```

```
f1.subs(x,10).subs(y,10).evalf(n=5)
✓ 0.0s
51.587
```

```
func = lambdify([x,y],f1,'numpy')
variables1 = np.arange(0,8);
variables2 = np.arange(5,13);
func(variables1,variables2)
✓ 0.1s
array([-1.          ,  0.35127873,  5.28171817, 13.51831093, 24.6109439 ,
        37.81750604, 51.91446308, 64.88454804])
```

```
type(f1)
```

✓ 0.1s

sympy.core.add.Add

```
f2 = 2*atan(x)-3*x+2
```

```
f2
```

✓ 0.0s

$-3x + 2 \operatorname{atan}(x) + 2$

```
simplify(f2)
```

✓ 0.0s

$-3x + 2 \operatorname{atan}(x) + 2$

```
f2.atoms()
```

✓ 0.3s

{-3, 2, x}

```
f2.evalf()
```

✓ 0.1s

$-3.0x + 2.0 \operatorname{atan}(x) + 2.0$

```
f2.subs(x,10).subs(y,10).evalf(n=5)
```

✓ 0.0s

-25.058

```
func = lambdify([x,y],f2,'numpy')
```

```
variables1 = np.arange(0,8);
```

```
variables2 = np.arange(5,13);
```

```
func(variables1,variables2)
```

✓ 0.0s

array([[2. , 0.57079633, -1.78570256, -4.50190846,
 -7.34836467, -10.25319847, -13.1887047 , -16.14220146]])

```
type(f2)
```

```
✓ 0.0s
```

```
sympy.core.add.Add
```