

```
# z01
"""
Найти в аналитическом виде сумму из n членов последовательности
нечетных чисел, начиная от 1.
"""
import sympy as sp

i, n, m = sp.symbols("i n m", integer=True)
res = sp.summation(2*i - 1, (i, 1, n))
print("res =", res)
```

```
# z02
"""
Найти сумму членов бесконечной последовательности
 $1/3 + 1/3^2 + 1/3^3 + 1/3^4 + \dots$ 
a) в аналитическом виде с исп. sympy.summation;
b) в численном виде с исп. sympy.mpmath.nsum;
c) в численном виде с исп. while (машинная точность).
"""
# a
import sympy as sp
"""
бесконечность - sympy.mpmath.inf
"""
oo = sp.oo # бесконечность (sp.oo эквивалентно sp.mpmath.inf)
i, n, m = sp.symbols("i n m", integer=True)
resa = sp.summation(1/3**i, (i, 1, oo))
print("resa =", resa)

# b
nsum(lambda n: 1/n**2, [1, inf])
resb = sp.mpmath.nsum(lambda n: 1/3**n, [1, sp.mpmath.inf])
print("resb =", resb)

# c
s = 0.0 # обнуление сумматора
sn = 1.0
x = 1.0/3.0
while 1:
    sn *= x # текущий член ряда
    ss = s + sn
    if s == ss: break
    s = ss
print("resc =", s)
```

```
# z03
"""
Найти предел последовательности
 $(1 + 1/n)**n, n=1,2,3,\dots$ 

Syntax: limit(function, variable, point)
"""
import sympy as sp
n = sp.symbols("n", integer=True)
res = sp.limit((1 + 1/n)**n, n, sp.oo)
print("res =", res)
```

```

# z04
"""
Найти предел  $x^x$  при  $x \rightarrow 0$ 

Syntax: limit(function, variable, point)
"""
import sympy as sp
x = sp.symbols("x", real=True)
res = sp.limit(x**x, x, 0)
print("res =", res)

```

```

# z05
"""
Найти предел  $\ln(1+\sin(4x))/(\exp(\sin(5x))-1)$  при  $x \rightarrow 0$ 

sympy.mpmath.log(x, b)
Computes the base-b logarithm of x, logb(x).
If b is unspecified, log() (page 561) computes
the natural (base e) logarithm and is equivalent to ln()
"""
import sympy as sp
x = sp.symbols("x", real=True)

def w(x):
    return sp.ln(1+sp.sin(4*x))/(sp.exp(sp.sin(5*x))-1)

res = sp.limit(w(x), x, 0)
print("res =", res)

```

```

# z06
"""
Найти предел  $n/(n!)^{1/n}$  при  $n \rightarrow \infty$ 
"""
import sympy as sp
n = sp.symbols("n", integer=True)

res = sp.limit(n/sp.factorial(n)**(1/n), n, sp.oo)
print("res =", res)

```

```

# z07
"""
Найти предел функции
 $\ln(1 + \sin(4x))/(\exp(\sin(5x))-1)$  при  $x \rightarrow 0$ 

log()
mpmath.log(x, b)
Computes the base-b logarithm of x, log_b(x).
If b is unspecified, log() computes the natural
(base e) logarithm and is equivalent to ln()
"""
from sympy import Symbol, ln, sin, exp, limit
x = Symbol("x", real=True)

f = lambda x: ln(1 + sin(4*x))/(exp(sin(5*x))-1)

res = limit(f(x), x, 0)
print("res =", res)

```

```
# z08
"""
Определить, имеет ли функция  $|2x - 3|/(2x - 3)$  скачок
при  $x = 3/2$ . Если имеет, то чему он равен.
"""
from sympy import oo, Symbol, Abs, limit
x = Symbol("x", real=True)

f = lambda x: Abs(2*x - 3)/(2*x - 3)

res1 = limit(f(x), x, 3/2, dir="+")
res2 = limit(f(x), x, 3/2, dir="-")
print("res1 =",res1," res2 =",res2," res =",res1-res2)
```

```
# z09
"""
Вывести численные значения числа Пифагора и
основания натурального алгоритма
"""
from sympy import pi,E
# x = Symbol("x", real=True)
res1 = pi.evalf()
res2 = E.evalf()
print("res1 =",res1," res2 =",res2)
```

```
# z10
"""
Раскрыть скобки  $(x+y)^3$ 
"""
from sympy import *
x, y = symbols("x y")
res = expand((x+y)**3)
print("res =",res)
```

```
# z11
"""
Раскрыть скобки  $(x+y)^3$ 
"""
from sympy import *
x, y = symbols("x y")
res = expand(cos(3*x), trig=True)
print("res =",res)
```

```
# z12
"""
Раскрыть скобки  $(x+y)^3$ 
"""
from sympy import *
x, y = symbols("x y")
res = expand((x+y)**3)
print("res =",res)
```

```
# z13
"""
Упростить выражение
 $\cos(x)\cos(y) - \sin(x)\sin(y)$  с помощью simplify
"""
from sympy import *
x, y = symbols("x y")
res = simplify(cos(x)*cos(y) - sin(x)*sin(y))
print("res =",res)
```

```
# z14
"""
Найти производную функции  $\sin(x)\ln(x)$ .
"""
from sympy import *
x = Symbol("x")
res = diff(sin(x)*ln(x), x)
print("res =",res)
```

```
# z15
"""
Найти 3-ю производную функции  $x**x$ .
"""
from sympy import *
x = Symbol("x")
res = diff(x**x, x, 3)
print("res =",res)
```

```
# z16
"""
Найти численное значение производной функции  $\sin(x + \pi/5)^2$ 
при  $x=1$ .

N() is equivalent to evalf()
"""
from sympy import *
x = Symbol("x")
res = diff(sin(x + pi/5)**2, x, 1).subs(x, 1).evalf()
print("res =",res)
```

```
# z17
"""
Разложить в ряд Тейлора функцию  $\sin(2x)/\cos(x)$  в окрестности
точки  $x=0$  до члена с  $x**11$  включительно.
"""
from sympy import *
x = Symbol("x")
f = sin(3*x)/cos(x)
res = f.series(x, 0, 12)
print("res =",res)
```

```

# z18
"""
Найти неопределённый интеграл от  $\sin(3x)$ .
Выполнить проверку.
"""
from sympy import *
x = Symbol("x")
f = lambda x: sin(3*x)
res = integrate(f(x), x)
print("res =", res)
# проверка
print(diff(res, x))

```

```

# z19
"""
Найти неопределённый интеграл от  $\exp(-x^2)\operatorname{erf}(x)$ .
"""
from sympy import *
x = Symbol("x")
f = lambda x: exp(-x**2)*erf(x)
res = integrate(f(x), x)
print("res =", res)
# проверка
print(diff(res, x))

```

```

# z20
"""
Найти определённый интеграл от  $\exp(-x^2)$ .
Пределы интегрирования от  $-\infty$  до  $+\infty$ .
"""
from sympy import *
x = Symbol("x")
res = integrate(exp(-x**2), (x, -oo, oo))
print("res =", res)

```

```

# z21
"""
Найти аналитическое решение ОДУ
 $f'(x) + f(x) = 0$ 
"""
from sympy import Function, Symbol, dsolve
f = Function('f')
x = Symbol('x')
res = dsolve(f(x).diff(x, x) + f(x), f(x))
print("res =", res)

```

```

# z22
"""
Найти аналитическое решение алгебраического уравнения
 $x^4 = 1$ .
"""
from sympy import solve, Symbol
x = Symbol("x")
res = solve(x**4 - 1, x)
print("res =", res)

```

```
# z23
"""
Найти аналитическое решение системы алгебраических уравнений
 $x + 5y - 2 = 0$ ;
 $3x - 6y + 15 = 0$ .
"""
from sympy import solve, symbols
x, y = symbols("x,y")
res = solve([x + 5*y - 2, 3*x - 6*y + 15], [x, y])
print("res =", res)
```

```
# z24
"""
Построить графики функций  $x \cdot \sin(x)$  и  $\sin(3x)/x$  ( $x \in [-6, +6]$ )
на одних координатных осях.
"""
import numpy as np
import matplotlib.pyplot as plt

ff = lambda xx: 3.0 if abs(xx) < 1.e-12 else np.sin(3*xx)/xx

nt = 101
x = np.linspace(-6.0, 6.0, nt)
y1 = x*np.sin(x)
y2 = [ff(x[i]) for i in range(nt)]

# plt.fill(x, y1, 'r')
plt.plot(x, y1, 'r', linewidth = 3.0, label = "y1(x)")
plt.plot(x, y2, 'b', linewidth = 3.0, label = "y2(x)" )
plt.grid(True)
plt.xlabel("X", fontsize = 16, color = "k")
plt.ylabel("Y1(X), Y2(X)", fontsize = 16, color = "k")
plt.legend(fontsize = 16)
plt.savefig("graph.pdf")
plt.show()
```