Exp1: To use mathematical software as an advance calculator

Name: Soniya Eknath Girhe.

Section: A7-B3

Roll Number: 59

Date: 21/01/2026.

In [1]:
```
585+15
```
Out[1]: 600

In [2]:
```
585-85
```
Out[2]: 500

In [3]:
```
50*50
```
Out[3]: 2500

In [4]:
```
59845/596
```
Out[4]: 59845/596

In [5]:
```
26/13
```
Out[5]: 2

In [6]:
```
5441/8
```
Out[6]: 5441/8

In [7]:
```
1548.0/4
```
Out[7]: 387.000000000000

In [8]:
```
4541/15.0
```
Out[8]: 302.733333333333

In [10]:
```
10/5.n() # .n() is for answer in decimal
```
Out[10]: 2.00000000000000

In [11]:
```
10//4 # // is for Quotient
```

Out[11]: 2

In [57]:
```
13/4.n(digits=7)
```

Out[57]: 3.250000

In [12]:
```
10%4 # % is for Remainder
```

Out[12]: 2

In [59]:
```
2^3
```

Out[59]: 8

In [60]:
```
2**5 # ** gives raise to power by
```

Out[60]: 32

In [61]:
```
A=factorial(10)
```

In [62]:
```
A
```

Out[62]: 3628800

In [65]:
```
A.ndigits() # .ndigits gives number of digits present
```

Out[65]: 7

In [66]:
```
A.digits() # .digits() gives list of all digits of A
```

Out[66]: [0, 0, 8, 8, 2, 6, 3]

In [70]:
```
C=A.digits()
C
```

Out[70]: [0, 0, 8, 8, 2, 6, 3]

In [69]:
```
C[1]
```

Out[69]: 0

In [73]:
```
C[3]
```

Out[73]: 8

In [74]:
```
13.is_prime()
```

Out[74]: True

In [75]:
```
12.is_prime()
```

Out[75]: False

In [77]:
```
12.next_prime()
```

Out[77]: 13

In [78]:
```
121.next_prime()
```

Out[78]: 127

In [79]:
```
list(primes(1,28))
```

Out[79]: [2, 3, 5, 7, 11, 13, 17, 19, 23]

In [83]:
```
exp(0)
```

Out[83]: 1

In [86]:
```
exp(oo)
```

Out[86]: +Infinity

In [85]:
```
exp(-infinity)
```

Out[85]: 0

Defining a Variable

In [13]:
```
A=1597865
B=1498959
```

In [15]:
```
A+B
```

Out[15]: 3096824

In [16]:
```
A-B
```

Out[16]: 98906

In [17]:
```
A/B
```

Out[17]: 1597865/1498959

In [18]:
```
A*B
```

Out[18]: 2395134122535

In [19]:
```
A/B.n()
```

Out[19]: 1.06598312562252

In [20]:
```
A.binary()
```

Out[20]: '110000110000110101001'

In [23]:
```
A.divisors()
```

Out[23]:
```
[1, 5, 313, 1021, 1565, 5105, 319573, 1597865]
```

In [1]:
```
factorial(5)
```

Out[1]:
```
120
```

Defining a Functions

In [27]:
```
f(x)=x^2+2*x
```

In [28]:
```
f.derivative()
```

Out[28]:
```
x |--> 2*x + 2
```

In [29]:
```
show(f.derivative())
```

In [37]:
```
show(f.integral(x))
```

In [34]:
```
integral?
```

```
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/models.py:84
7: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:67: De
precationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:80: De
precationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:93: De
precationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:108: D
eprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:122: D
eprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:136: D
eprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/api.py:148: D
eprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:4
98: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:5
09: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:5
20: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:5
33: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:5
44: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:5
55: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/requests/sessions.py:5
65: DeprecationWarning: invalid escape sequence \*
  """
/opt/sagemath-9.3/local/lib/python3.7/site-packages/docutils/writers/latex
2e/__init__.py:2978: DeprecationWarning: invalid escape sequence \l
  self.out.append('}] \leavevmode ')
```

In [38]:
```
show(f.integral(x, 1,2))
```

In [39]:
```
sin(pi/2)
```

Out[39]: 1

In [40]:
```
cos(pi/2)
```

Out[40]: 0

In [41]:
```
tan(pi/2)
```

Out[41]: Infinity

In [44]:
```
show(asin(1))
```

In [45]:
```
show(acos(0))
```

In [46]:
```
show(atan(oo)) # Write oo for Infinity
```

In [48]:
```
log(10.0)
```

Out[48]: 2.30258509299405

In [49]:
```
log(10, 10)
```

Out[49]: 1

In [50]:
```
gcd(10, 15)
```

Out[50]: 5

In [51]:
```
lcm(10, 15)
```

Out[51]: 30

Definig the Formula

Find the Area of Triangle whose sides are 3, 4, 5 by definig a formula

In [52]:
```
def Heron(a, b, c):
    s=(a+b+c)/2.0
    Area=sqrt(s*(s-a)*(s-b)*(s-c))
    return(Area)
```

In [53]:
```
Heron(3, 4, 5)
```

Out[53]: 6.00000000000000

In [54]:
```
Heron(13, 14, 15)
```

Out[54]: 84.0000000000000

Find the Sum of first 10 Natural Number

In [55]:
```
def SumN(n):
    N=n*(n+1)/2.0
    return(N)
```

In [56]:
```
SumN(10)
```

Out[56]: 55.0000000000000

In [57]:
```
SumN(50)
```

Out[57]: 1275.00000000000

In [58]:
```
SumN(15)
```

Out[58]: 120.000000000000

Find the Sum of first 10 Natural Number by using For loop

In [64]:
```
k=0
for i in range (1,10):
    k=k+i
k
## in python 10 will exclude therfor put 11 instede 10
```

Out[64]: 45

In [63]:
```
k=0
for i in range (1,11):
    k=k+i
k
```

Out[63]: 55

Solve Simultaneous Equations $x+y=10 and x-y=2$

In [69]:
```
x,y=var('x,y')
show(solve([x+y==10,x-y==2],x,y))
```

In [5]:
```
f(x)=x^2-6*x+10
```

In [71]:
```
show(f.roots())
```

In [72]:
```
f(x)=x^3-2*x^2-5*x+6
```

In [74]:
```
show(f.roots()) # 1 is multiplexicity
```

Exercise Problem

1. Find the roots of $x^3 − 2x^2 − 5x + 6 = 0.$

In [14]:
```
f(x)=x^3-2*x^2-5*x+6
```

In [16]:
```
show(f.roots())
```

2. Solve the system of non linear equations $x^2+y^2=4$ and $y=x^2-2$ for x and y.

In [17]:
```
x,y=var('x,y')
show(solve([x^2+y^2==4,y==x^2-2],x,y))
```

3. Find the factors of sum of digits of 275!.

In [19]:
```
show(factorial(275))
```

4. Suppose an investment is made to a bank by an individual. The bank gives an annual interest at the rate 5%. Return is calculated by using compound interest. Create an user defined function to input the investment amount, the number of years for which investment is made, and print the returns.

In [30]:
```
def calculate_returns(principal, years):
    rate = 0.05
    growth = 1
    for i in range(years):
        growth = growth * (1 + rate)
    amount = principal * growth
    return amount
```

In [31]:
```
calculate_returns(10000, 5)
```

Out[31]:
```
12762.8156250000
```

5. Write a code for finding the area of a quadrilateral by defining the formulaes.

In [32]:
```
def quadrilateral_area(a, b, c, d):
    # semi-perimeter
    s = (a + b + c + d) / 2
    # formula
    area = sqrt((s - a) * (s - b) * (s - c) * (s - d))
    return area
```

In [35]:
```
show(quadrilateral_area(5, 6, 7, 8))
```

In [36]:
```
show(quadrilateral_area(5.0, 6, 7, 8))
```

6. Write a formulae to derive the sum of any AP series where initial term, common differences and number of terms is given.

In [39]:
```
def ap_sum(a, d, n):
    sum_ap = (n / 2) * (2 * a + (n - 1) * d)
    return sum_ap
```

In [38]:
```
ap_sum(2, 3, 5)
```

Out[38]: 40

7. Define a function for finding the mean of a list of numbers.

In [47]:
```
def find_mean(numbers):
    # Step 1: calculate total sum
    total = 0
    for num in numbers:
        total = total + num

    # Step 2: count elements manually
    count = 0
    for num in numbers:
        count = count + 1

    # Step 3: mean formula
    mean = total / count
    return mean
```

In [48]:
```
find_mean([10, 20, 30, 40, 50])
```

Out[48]: 30

8. Define a function for finding factorial of a number.

In [49]:
```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result = result * i
    return result
```

In [50]:
```
factorial(5)
```

Out[50]:
```
120
```

9. Define a function to check whether a given number is prime or not?

In [51]:
```
def is_prime(n):
    # Prime numbers are greater than 1
    if n <= 1:
        return False

    # Check divisibility from 2 up to n-1
    for i in range(2, n):
        if n % i == 0:   # if divisible
            return False
    return True
```

In [52]:
```
print(is_prime(7))   # True
print(is_prime(10))  # False
```

```
True
False
```

10. Define a formulae to find the sum of square of first natural numbers.

In [55]:
```
def sum_of_squares(n):
    return (n * (n + 1) * (2 * n + 1)) / 6
```

In [56]:
```
sum_of_squares(5)
```

Out[56]:
```
55
```

Learning: SageMath as an Advanced Calculator

The objective of this experiment was to utilize mathematical software to perform complex calculations and automate mathematical formulas. The key learnings include:

In this first lab experiment, I explored how to use SageMath as a powerful computational tool for engineering mathematics. Here is a summary of what I learned:

Precision and Number Theory: I learned to handle both symbolic and numerical calculations, as well as tools for prime testing, factorials, and finding GCD/LCM. Large Data Handling: I practiced analyzing massive integers by counting their digits and using indexing to access specific values within a large number. Algebraic Solvers: I discovered how to define variables and solve complex equations and systems of non-linear equations automatically. Programming Logic: I learned to create user-defined functions and use loops to automate repetitive mathematical formulas and series calculations.