Digital SAT Practice 1 (Mathematics Section)

Importing Libraries

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Questions

Module 1

1. What is 10% of 470?

In [ ]:
```
x = 470
percent = 10

result = x * percent / 100
print(result)
```

```
47.0
```

2. 4x + 6 = 18, Find 4x

In [ ]:
```
x = (18 - 6) / 4
print(4*x)
```

```
12.0
```

4. The function g is defined by g(x) = x^2 + 9. For which value of x is g(x) = 25?

In [ ]:
```
x = np.sqrt(25 - 9)
print(x)
```

```
4.0
```

6. A printer produces posters at a constant rate of 42 posters per minute. At what rate, in posters per **hour** does the printer produce the posters?

In [ ]:
```
rate = 42 * 60 # 42 posters per minute * 60 minutes per hour
print(rate)
```

```
2520
```

7. The function f is defined by the equation f(x) = 7x + 2. What is the value of f(x) when x = 4?

In [ ]:
```
# Define the function f(x)
def f(x):
    return 7 * x + 2

# Define the value of x
x = 4

# Calculate the value of f(x) when x = 4
result = f(x)

# Print the result
print("The value of f(x) when x = 4 is:", result)
```

```
The value of f(x) when x = 4 is: 30
```

9. Right triangles LMN and PQR are similar, where L and M correspond to P and Q, respectively. Angle M has a measure of 53 deg. What is the measure of angle Q?

In [ ]:
```
# Measure of angle M
angle_M = 53  # degrees

# Measure of angle Q (since angles M and Q are corresponding angles in similar triangles)
angle_Q = angle_M

print("Measure of angle Q:", angle_Q, "degrees")
```

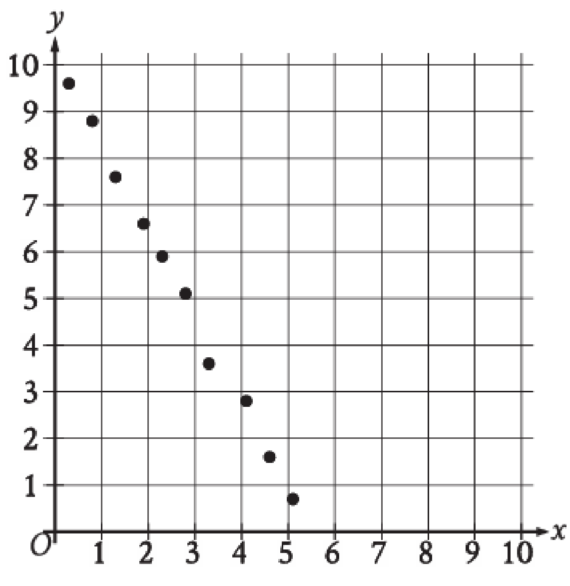
```
Measure of angle Q: 53 degrees
```

10. y = -3x and 4x+y = 15. The solution to the given system of equations is (x, y). What is the value of x?

In [ ]:
```
from sympy import symbols, Eq, solve

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

# Define the equations
equation1 = Eq(y, -3 * x)
equation2 = Eq(4 * x + y, 15)

# Solve the system of equations
solution = solve((equation1, equation2), (x, y))

# Extract the value of x from the solution
x_value = solution[x]

print("Value of x:", x_value)
```

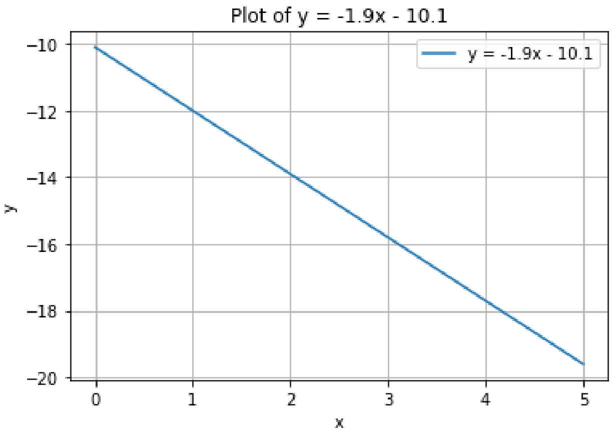
```
Value of x: 15
```

11. Which of the following equations is the most appropriate linear model for the data shown in the scatterplot?



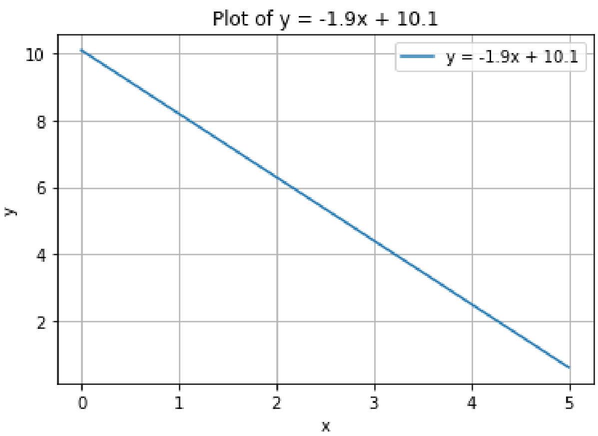
In [ ]:
```
# Generate x values
x_values = np.linspace(0, 5, 10)  # Generate 100 x values from -10 to 10

# Compute corresponding y values
y_values = -1.9 * x_values - 10.1  # Substitute x values into the equation

# Plot the line
plt.plot(x_values, y_values, label='y = -1.9x - 10.1')

# Add labels and title
plt.xlabel('x')
plt.ylabel('y')
```

```python
plt.title('Plot of y = -1.9x - 10.1')

# Add grid
plt.grid(True)

# Add legend
plt.legend()

# Show plot
plt.show()
```



In [ ]:

```python
# Generate x values
x_values = np.linspace(0,5, 10)  # Generate 100 x values from -10 to 10

# Compute corresponding y values
y_values = -1.9 * x_values + 10.1  # Substitute x values into the equation

# Plot the line
plt.plot(x_values, y_values, label='y = -1.9x + 10.1')

# Add labels and title
plt.xlabel('x')
plt.ylabel('y')
plt.title('Plot of y = -1.9x + 10.1')

# Add grid
plt.grid(True)

# Add legend
plt.legend()

# Show plot
plt.show()
```



13. Vivian bought party hats and cupcakes for $71.Eachpackageofpartyhatscost3$, and each cupcake cost $1. If Vivian bought 10 packages of party hats, how many cupcakes did she buy?

In [ ]:

```python
# Define the cost of each item
cost_per_hat = 3  # $3 per package of party hats
```

```
cost_per_cupcake = 1  # $1 per cupcake

# Define the total amount spent
total_spent = 71  # $71 in total

# Calculate the total cost of party hats
total_cost_hats = 10 * cost_per_hat  # 10 packages of party hats

# Calculate the remaining amount for cupcakes
remaining_amount = total_spent - total_cost_hats

# Calculate the number of cupcakes bought
num_cupcakes = remaining_amount / cost_per_cupcake

# Print the result
print("Vivian bought", num_cupcakes, "cupcakes.")
```

```
Vivian bought 41.0 cupcakes.
```

14. z^2 + 10z - 24 = 0, what is one of the solutions to the given equation?

In [ ]:
```
import cmath  # Import the complex math module

# Coefficients of the quadratic equation
a = 1
b = 10
c = -24

# Calculate the discriminant
discriminant = (b ** 2) - (4 * a * c)

# Calculate the solutions using the quadratic formula
solution1 = (-b - cmath.sqrt(discriminant)) / (2 * a)
solution2 = (-b + cmath.sqrt(discriminant)) / (2 * a)

# Print one of the solutions
print("One of the solutions to the equation is:", solution2)
```

```
One of the solutions to the equation is: (2+0j)
```

15. Bacteria are growing in a liquid growth medium. There were 300,000 cells per milliliter during an initial observation. The number of cells per milliliter doubles every 3 hours. How many cells per milliliter will there be 15 hours after the initial observation?

In [ ]:
```
# Initial number of cells per milliliter
initial_cells_per_ml = 300000

# Time after initial observation (in hours)
time_hours = 15

# Number of times the population doubles in the given time
doubling_times = time_hours / 3

# Calculate the final number of cells per milliliter
final_cells_per_ml = initial_cells_per_ml * (2 ** doubling_times)

# Print the result
print("After", time_hours, "hours, there will be approximately", final_cells_per_ml, "cells per millilit
```

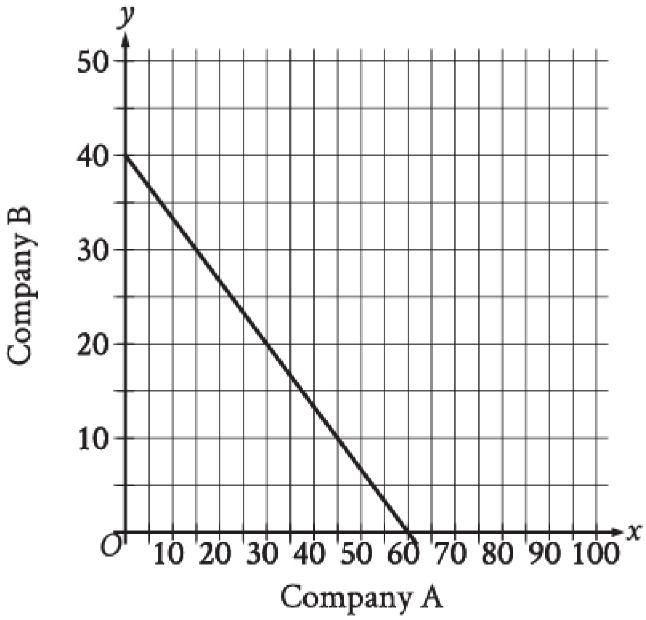
```
After 15 hours, there will be approximately 9600000.0 cells per milliliter.
```

16. Make the factorization on 6x^8 y^2 + 12x^2 y^2

In [ ]:
```
from sympy import symbols, factor

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

# Expression to factorize
expression = 6 * x**8 * y**2 + 12 * x**2 * y**2

# Factorize the expression
```

```
factorized_expression = factor(expression)

# Print the factorized expression
print("Factorized expression:", factorized_expression)
```

```
Factorized expression: 6*x**2*y**2*(x**6 + 2)
```



18.

The graph shows the relationship between the number of shares of stock from Company A, x, and the number of shares of stock from Company B, y, that Simone can purchase. Which equation could represent this relationship?

A. y = 8x + 12

B. 8x + 12y = 480

C. y = 12x + 8

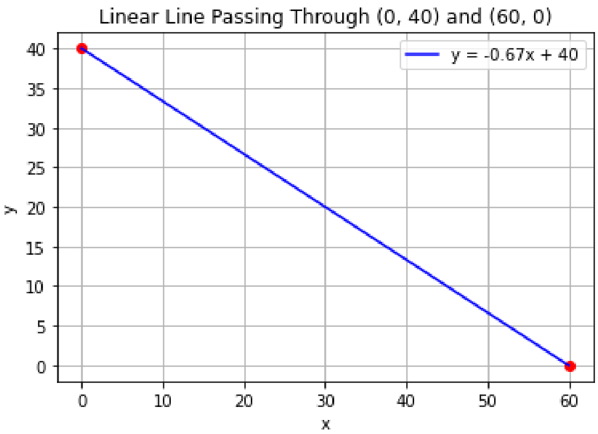
D. 12x + 8y = 480

In [ ]:
```
# Given points
x1, y1 = 0, 40
x2, y2 = 60, 0

# Calculate the slope (m)
slope = (y2 - y1) / (x2 - x1)

# Choose one point to substitute into the point-slope form (let's choose the first point)
x_point, y_point = x1, y1

# Use point-slope form to find the equation of the line
# y - y1 = m(x - x1)
# => y - y_point = slope * (x - x_point)
# => y = slope * (x - x_point) + y_point
equation = f"y = {slope:.2f}(x - {x_point}) + {y_point}"

# Print the equation of the line
print("Equation of the line:", equation)
```

```
Equation of the line: y = -0.67(x - 0) + 40
```

In [ ]:
```
import matplotlib.pyplot as plt

# Given points
x1, y1 = 0, 40
x2, y2 = 60, 0

# Calculate the slope (m)
slope = (y2 - y1) / (x2 - x1)

# Generate x values for the line
x_values = [x1, x2]
```

```python
# Calculate corresponding y values for the line using the equation of the line
y_values = [y1, y2]

# Plot the points
plt.plot([x1, x2], [y1, y2], 'ro')  # 'ro' for red circles

# Plot the line
plt.plot(x_values, y_values, label=f'y = {slope:.2f}x + {y1}', color='blue')

# Add labels and title
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Line Passing Through (0, 40) and (60, 0)')

# Add grid
plt.grid(True)

# Add legend
plt.legend()

# Show plot
plt.show()
```



19. Circle A has a radius of 3n and circle B has a radius of 129n, where n is a positive constant. The area of circle B is how many times the area of circle A.

In [ ]:
```python
import math

# Radius of circle A and circle B
radius_A = 3 * x
radius_B = 129 * x

# Calculate the area of circle A
area_A = math.pi * (radius_A ** 2)

# Calculate the area of circle B
area_B = math.pi * (radius_B ** 2)

# Calculate how many times the area of circle B is compared to the area of circle A
times_area_B_to_A = area_B / area_A

# Print the result
print("The area of circle B is", times_area_B_to_A, "times the area of circle A.")
```

```
The area of circle B is 1849.00000000000 times the area of circle A.
times the area of circle A.
```

20. The frequency table summarizes the 57 data values in a data set. What is the maximum data value in the data set?

Data Value	Frequency
6	3
7	3

Data Value	Frequency
8	8
9	8
10	9
11	11
12	9
13	0
14	6

In [ ]:
```
# Define the frequency table
frequency_table = {
    6: 3,
    7: 3,
    8: 8,
    9: 8,
    10: 9,
    11: 11,
    12: 9,
    13: 0,
    14: 6
}

# Find the maximum data value
max_data_value = max(frequency_table.keys())

# Print the result
print("The maximum data value in the data set is:", max_data_value)
```

```
The maximum data value in the data set is: 14
```

21. A circle in the xy-plane has diameter with endpoints (2, 4) and (2, 14). An equation of this circle is (x-2)^2 + (y-9)^2 = r^2, where r is a positive constant. What is the value of r?

In [ ]:
```
# Endpoints of the diameter
x1, y1 = 2, 4
x2, y2 = 2, 14

# Calculate the midpoint (x-coordinate of the center is the same as the x-coordinate of the endpoints)
center_x = x1

# The y-coordinate of the center is the average of the y-coordinates of the endpoints
center_y = (y1 + y2) / 2

# Calculate the radius using the distance formula: sqrt((x2 - x1)^2 + (y2 - y1)^2)
radius = ((x2 - center_x)**2 + (y2 - center_y)**2) ** 0.5

# Print the value of r (radius)
print("The value of r (radius) is:", radius)
```

```
The value of r (radius) is: 5.0
 5.0
```

22. The measure of angle R is (2pi)/3 rad. The measure of angle T is (5pi)/12 rad greater than the measure of angle R. What is the measure of angle T, in degrees?

In [ ]:
```
import math

# Measure of angle R in radians
angle_R_rad = 2 * math.pi / 3

# Measure of angle T in radians greater than angle R
angle_T_rad = angle_R_rad + (5 * math.pi / 12)

# Convert radians to degrees for both angles
angle_R_deg = math.degrees(angle_R_rad)
angle_T_deg = math.degrees(angle_T_rad)
```

```
# Print the measure of angle T in degrees
print("The measure of angle T is:", angle_T_deg, "degrees.")
```

```
The measure of angle T is: 194.99999999999997 degrees.
```

23. A certain town has an area of 4.36 square miles. What is the area, in the **square yards**, of this town? (1 mile = 1760 yards)

In [ ]:
```
# Area in square miles
area_miles = 4.36

# Conversion factor from square miles to square yards
conversion_factor = 1760 ** 2  # 1 mile = 1760 yards, so 1 square mile = (1760 yards)^2

# Convert area from square miles to square yards
area_yards = area_miles * conversion_factor

# Print the area in square yards
print("The area of the town in square yards is:", area_yards, "square yards.")
```

```
The area of the town in square yards is: 13505536.000000002 square yards.
```

24.

x	y
18	130
23	160
26	178

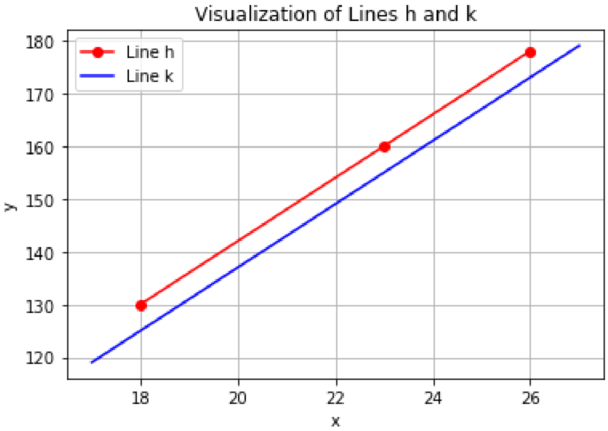
For line h, the table shows three values of x and their corresponding values of y. Line k is the result of translating line h down 5 units in the xy-plane. What is the x-intercept of line k?

In [ ]:
```
# Given points for line h
x_h = [18, 23, 26]
y_h = [130, 160, 178]

# Calculate the slope of line h
slope_h = (y_h[1] - y_h[0]) / (x_h[1] - x_h[0])

# Calculate the y-intercept of line h using one of the points
y_intercept_h = y_h[0] - slope_h * x_h[0]

# Translate line h down 5 units to get the equation of line k
y_intercept_k = y_intercept_h - 5

# Calculate the x-intercept of line k (where y = 0)
x_intercept_k = -y_intercept_k / slope_h

# Print the x-intercept of line k
print("The x-intercept of line k is:", x_intercept_k)
```

```
The x-intercept of line k is: -2.8333333333333335
```

In [ ]:
```
import matplotlib.pyplot as plt

# Given points for line h
x_h = [18, 23, 26]
y_h = [130, 160, 178]

# Calculate the slope of line h
slope_h = (y_h[1] - y_h[0]) / (x_h[1] - x_h[0])

# Calculate the y-intercept of line h using one of the points
y_intercept_h = y_h[0] - slope_h * x_h[0]

# Translate line h down 5 units to get the equation of line k
y_intercept_k = y_intercept_h - 5

# Calculate the x-intercept of line k (where y = 0)
```

```python
x_intercept_k = -y_intercept_k / slope_h

# Generate x values for lines h and k
x_values = [min(x_h) - 1, max(x_h) + 1]

# Calculate corresponding y values for lines h and k
y_values_h = [slope_h * x + y_intercept_h for x in x_values]
y_values_k = [slope_h * x + y_intercept_k for x in x_values]

# Plot the points and lines
plt.plot(x_h, y_h, 'ro-', label='Line h')
plt.plot(x_values, y_values_k, 'b-', label='Line k')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Visualization of Lines h and k')
plt.legend()
plt.grid(True)
plt.show()
```

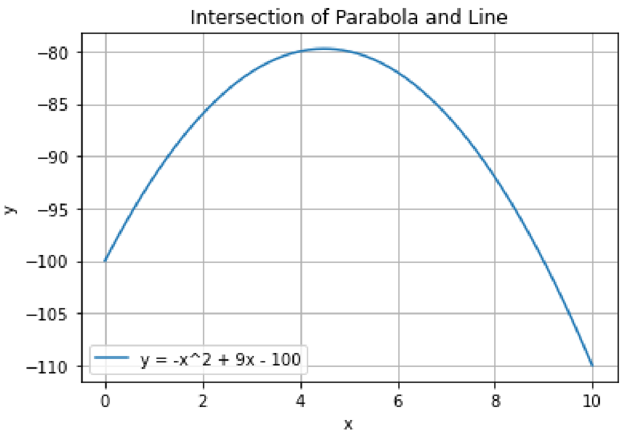


25. In the xy-plane, the graph of the equation y = -x^2 + 9x - 100 intersects the line y = c at exactly one point. What is the value of c?

In [ ]:

```python
import numpy as np
import matplotlib.pyplot as plt

# Define the range of x values
x_values = np.linspace(0, 10, 1000)

# Define the equations of the parabola and the line
def parabola_equation(x):
    return -x**2 + 9*x - 100

def line_equation(c):
    return c

# Find the x-coordinate of the intersection point by setting the equations equal to each other
def find_intersection():
    for x in x_values:
        if parabola_equation(x) == line_equation(c):
            return x

# Define possible values of c
c_values = np.linspace(-200, 200, 1000)

# Find the x-coordinate of the intersection point for each value of c
intersection_points = []
for c in c_values:
    x_intersection = find_intersection()
    if x_intersection is not None:
        intersection_points.append((x_intersection, c))

# Plot the parabola
plt.plot(x_values, parabola_equation(x_values), label='y = -x^2 + 9x - 100')

# Plot the lines corresponding to each value of c
for point in intersection_points:
```

```
    plt.axhline(y=point[1], color='r', linestyle='--', label=f'y = {point[1]}')

# Plot the intersection points
for point in intersection_points:
    plt.plot(point[0], point[1], 'ro')  # 'ro' for red circle marker
# Add labels, title, legend, and grid
plt.xlabel('x')
plt.ylabel('y')
plt.title('Intersection of Parabola and Line')
plt.legend()
plt.grid(True)

# Show plot
plt.show()
```



27. The perimeter of an equilateral triangle is 624 centimeters. The height of this triangle is k√(3) centimeters, where k is a constant. What is the value of k?

In [ ]:
```
import math

# Perimeter of the equilateral triangle
perimeter = 624  # centimeters

# Formula to find the side length of the equilateral triangle
side_length = perimeter / 3

# Height of the equilateral triangle
height = side_length * math.sqrt(3) / 2

# Value of k
k = height / math.sqrt(3)

# Print the value of k
print("The value of k is:", k)
```

```
The value of k is: 104.0
```