def long\_multiplication(num1, num2):

# Convert numbers to strings to iterate through digits

num1\_str = str(num1)

num2\_str = str(num2)

# Initialize result with zeros

result = [0] \* (len(num1\_str) + len(num2\_str))

# Multiply each digit and store the result in the appropriate position

steps = []

for i in range(len(num1\_str) - 1, -1, -1):

for j in range(len(num2\_str) - 1, -1, -1):

# Calculate the product of digits and add it to the result array

mul = int(num1\_str[i]) \* int(num2\_str[j])

pos1 = i + j

pos2 = i + j + 1

total = mul + result[pos2]

result[pos1] += total // 10

result[pos2] = total % 10

# Save step-by-step calculation

step = f"{num1\_str[i]} \* {num2\_str[j]} = {mul} -> "

step += f"Add {total // 10} to {result[pos1]} and put {total % 10} in position {pos2}"

steps.append(step)

# Convert result list to a string

result\_str = ''.join(map(str, result)).lstrip('0')

# Check if the result is an empty string (occurs when result is 0)

if not result\_str:

return "0", steps

return result\_str, steps

# Get user input for two numbers

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

# Perform long multiplication

result, multiplication\_steps = long\_multiplication(num1, num2)

# Display the step-by-step method

print(f"The result of {num1} \* {num2} is: {result}\n")

print("Step-by-step method:")

for step in multiplication\_steps:

print(step)

def long\_multiplication(num1, num2):

num1\_str = str(num1)

num2\_str = str(num2)

result = [0] \* (len(num1\_str) + len(num2\_str))

steps = []

for i in range(len(num1\_str) - 1, -1, -1):

for j in range(len(num2\_str) - 1, -1, -1):

mul = int(num1\_str[i]) \* int(num2\_str[j])

pos1 = i + j

pos2 = i + j + 1

total = mul + result[pos2]

result[pos1] += total // 10

result[pos2] = total % 10

step = f"{num1\_str[i]} \* {num2\_str[j]} = {mul} -> "

step += f"Add {total // 10} to {result[pos1]} and put {total % 10} in position {pos2}"

steps.append(step)

result\_str = ''.join(map(str, result)).lstrip('0')

if not result\_str:

return "0", steps

return result\_str, steps

def visualize\_multiplication(num1, num2):

result, multiplication\_steps = long\_multiplication(num1, num2)

print(f"The result of {num1} \* {num2} is: {result}\n")

print("Step-by-step method:")

print(f"{'-' \* 40}")

print(f"| {'Multiplication':^15} | {'Addition':^23} |")

print(f"{'-' \* 40}")

for step in multiplication\_steps:

mul, add = step.split(" -> ")

mul = mul.split("=")[1].strip()

add = add.split(" and put ")

add\_first = add[0].split("Add ")[1]

add\_second = add[1].split(" in position ")

print(f"| {mul:^15} | {add\_first + ' ':^15} | {add\_second[1]:^15} |")

print(f"{'-' \* 40}")

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

visualize\_multiplication(num1, num2)

def perform\_arithmetic\_operation(operation, num1, num2):

if operation == '+':

return num1 + num2

elif operation == '-':

return num1 - num2

elif operation == '\*':

return num1 \* num2

elif operation == '/':

if num2 != 0:

return num1 / num2

else:

return "Cannot divide by zero"

else:

return "Invalid operation"

def calculate\_percentage(original\_value, percentage):

return (percentage / 100) \* original\_value

def solve\_linear\_equation(a, b):

if a != 0:

return -b / a

else:

return "The equation is not linear (a cannot be zero)"

# Example usage:

print("GCSE Math Problem Solver")

print("========================\n")

print("1. Arithmetic Operations:")

num1 = float(input("Enter the first number: "))

operation = input("Enter the operation (+, -, \*, /): ")

num2 = float(input("Enter the second number: "))

result = perform\_arithmetic\_operation(operation, num1, num2)

print(f"Result of {num1} {operation} {num2} is: {result}\n")

print("2. Percentage Calculation:")

original\_value = float(input("Enter the original value: "))

percentage = float(input("Enter the percentage (without % symbol): "))

percentage\_result = calculate\_percentage(original\_value, percentage)

print(f"{percentage}% of {original\_value} is: {percentage\_result}\n")

print("3. Linear Equation Solver (ax + b = 0):")

a\_value = float(input("Enter the 'a' value: "))

b\_value = float(input("Enter the 'b' value: "))

solution = solve\_linear\_equation(a\_value, b\_value)

print(f"The solution to the equation {a\_value}x + {b\_value} = 0 is: {solution}")

23 \* 23 = 3 times 3 = 9

3 \* 20 (because it is in the 10 units) = 60

Then 20 (because it is in the 10 units) \*3 =60

After that 20 (because it is in the 10 units) \* 20 (because it is in the 10 units) = 400

Finally 400 + 60 + 60 + 9 = 529