# Software Engineering: Assignment 1

# “Looking for Fermat’s Last Theorem Near Misses”

## Source Code

# Function to calculate the relative miss

def calculate\_miss(x, y, z, n):

target = x\*\*n + y\*\*n

miss = min(abs(target - z\*\*n), abs((z+1)\*\*n - target))

RelativeMisses = (miss / target) \* 100

return miss, RelativeMisses

# Function to find the best near misses

def find\_nearestmisses(n, k):

# Initialize variables

Smallest\_RelativeMisses = float('inf')

bestX, bestY, bestZ = None, None, None

for x in range(10, k+1):

for y in range(10, k+1):

for z in range(1, k+1):

# Calculate the miss and relative miss

miss, RelativeMisses = calculate\_miss(x, y, z, n)

# Update best near miss

if RelativeMisses < Smallest\_RelativeMisses:

Smallest\_RelativeMisses = RelativeMisses

bestX, bestY, bestZ = x, y, z

return bestX, bestY, bestZ, Smallest\_RelativeMisses

# Main function

def main():

print("Welcome to Fermat's Last Theorem Near Misses\n")

# Get user input

n = int(input("Enter the n Value,n must be greater than 2 and less than 12:"))

k = int(input("\nEnter the K Value, must be greater than 10:"))

# Check if input values are valid

if 2 < n < 12 and k > 10:

bestX, bestY, bestZ, Smallest\_RelativeMisses = find\_nearestmisses(n, k)

# Display output

print("\nBest near miss:")

print(f"x: {bestX}, y: {bestY}, z: {bestZ}")

print(f"Actual miss: {calculate\_miss(bestX, bestY, bestZ, n)[0]}")

print(f"Relative miss: {Smallest\_RelativeMisses:.2f}%")

else:

print("Invalid input. Please make sure n is between 2 and 12, and k is greater than 10.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

## Output

