**Problem 1:** Use a direct proof to show that the sum of two odd integers is even.

**Answer:**

If x is an odd integer, then there exists an integer *y* such that *x* =2*y* + 1.

If x is an even integer, then there exists an integer *y* such that *x* = 2*y*.

Let *a* and *b* be odd integers, then there exists integers *y* and *z* such that:

*a* = 2*y* + 1

*b* = 2*z* + 1

We want the sum of the 2 odd integers:

a + b = 2y + 1 + 2z + 1 = 2y + 2z + 2 = 2(y + z + 1)

Since *y* and *z* are integers, *y* + *z* + 1 is also an integer and thus *a* + *b* is even.

**Problem 2:**  Show that if n and n3 + 5 is odd, then n is even using

1. a proof by contraposition
2. a proof by contradiction

**Answer:**

1. Proof by contraposition
   1. Assume n is odd
   2. There is an integer k
      1. n = 2k + 1
   3. Substituting
      1. n3 + 5 = (2k + 1)3 + 5 = 2(4k3 + 6k2 + 3k + 3)
      2. Find m = 4k3 + 6k2 + 3k + 3 so that
         1. n3 + 5 = 2m
   4. Means that n3 + 5 is even
   5. Since contraposition is true, then the original statement is true
2. Proof by contradiction
   1. Assume that n3 + 5 is odd and n is not even
   2. Because n is not even then n is odd
   3. So n = 2k + 1
   4. Substituting
      1. n3 + 5 = (2k+1)3 + 5 = 2(4k3 + 6k2 + 3k + 3)
      2. Find m = 4k3 + 6k2 + 3k + 3 so that
         1. n3 + 5 = 2m
   5. Means that n3 + 5 is even
   6. But, we assumed that it is both even and odd, so it’s a contradiction
   7. So “n is odd” is false

**Problem 3:**  Create a short Python program that allows you to prove that there are no positive perfect cubes less than 1000 that are the sum of the cubes of two positive integers.  Then,

1. Attach your code to this word document.
2. Write a formal proof using one of the techniques discussed in lecture.

**Code:**

def is\_perfect\_cube(number) -> bool:

"""

Indicates (with True/False) if the provided number is a perfect cube.

"""

number = abs(number) # Prevents errors due to negative numbers

return round(number \*\* (1 / 3)) \*\* 3 == number

def isPerfectCube(number):

    if round(number \*\* (1/3)) \*\* 3 == number:

        return number

# See if each number is a perfect cube and check whether a sum when cubed equals a cube

isSum = True

for i in range(1, 1000):

    is\_cube = is\_perfect\_cube(i)

    isCube = isPerfectCube(i)

    if is\_cube:

        print(i, "is a perfect cube!")

        for j in range(1, isCube):

            oneInt = isCube - j

            if (((oneInt \*\* 3) + (j \*\* 3)) == isCube):

                print(str(oneInt) + " and " + str(j) + " are sums to " + str(isCube))

            else:

                isSum = False

if not isSum:

    print("There are no two numbers when cubed equal to a cube.")

**Proof:**

Using Exhaustion:

13 = 1

23 = 8

33 = 27

43 = 64

53 = 125

63 = 216

73 = 343

83 = 512

93 = 729

103 = 1000

We want x3 + y3 = z3

So 1 + 1 = 2 ≠ 27

1 + 8 = 9 ≠ 27

8 + 8 = 16 ≠ 27

1 + 27 = 28 ≠ 64

8 + 27 = 35 ≠ 64

27 + 27 = 54 ≠ 64

1 + 64 = 65 ≠ 125

8 + 64 = 72 ≠ 125

27 + 64 = 91 ≠ 125

64 + 64 = 128 ≠ 125

1 + 125 = 126 ≠ 216

8 + 125 = 133 ≠ 216

27 + 125 = 152 ≠ 216

64 + 125 = 189 ≠ 216

125 + 125 = 250 ≠ 216

1 + 216 = 217 ≠ 343

8 + 216 = 224 ≠ 343

27 + 216 = 243 ≠ 343

64 + 216 = 280 ≠ 343

125 + 216 = 341 ≠ 343

216 + 216 = 432 ≠ 343

1 + 343 = 344 ≠ 512

8 + 343 = 351 ≠ 512

27 + 343 = 370 ≠ 512

64 + 343 = 407 ≠ 512

125 + 343 = 468 ≠ 512

216 + 343 = 559 ≠ 512

343 + 343 = 686 ≠ 512

1 + 512 = 513 ≠ 729

8 + 512 = 520 ≠ 729

27 + 512 = 539 ≠ 729

64 + 512 = 576 ≠ 729

125 + 512 = 637 ≠ 729

216 + 512 = 728 ≠ 729

343 + 512 = 855 ≠ 729

512 + 512 = 1024 ≠ 729

By exhaustion (literally, lol), we’ve shown that there are no positive perfect cubes less than 1000 that are the sum of the cubes of two positive integers.