#1,

Def number\_to\_words(num):

Def one(num):

Switch = [“”, “One”, “Two”, “Three”, “Four”, “Five”, “Six”, “Seven”, “Eight”, “Nine”]

Return switch[num]

Def two\_less\_20(num):

Switch = [“Ten”, “Eleven”, “Twelve”, “Thirteen”, “Fourteen”, “Fifteen”, “Sixteen”, “Seventeen”, “Eighteen”, “Nineteen”]

Return switch[num – 10]

Def tens(num):

Switch = [“”, “”, “Twenty”, “Thirty”, “Forty”, “Fifty”, “Sixty”, “Seventy”, “Eighty”, “Ninety”]

Return switch[num]

Def two(num):

If num == 0:

Return “”

Elif num < 10:

Return one(num)

Elif num < 20:

Return two\_less\_20(num)

Else:

Return tens(num // 10) + (“ “ + one(num % 10) if num % 10 != 0 else “”)

Def three(num):

If num < 100:

Return two(num)

Else:

Return one(num // 100) + “ Hundred” + (“ “ + two(num % 100) if num % 100 != 0 else “”)

If num == 0:

Return “Zero”

Billion = num // 1\_000\_000\_000

Million = (num % 1\_000\_000\_000) // 1\_000\_000

Thousand = (num % 1\_000\_000) // 1\_000

Remainder = num % 1\_000

Result = “”

If billion:

Result += three(billion) + “ Billion”

If million:

Result += “ “ if result else “”

Result += three(million) + “ Million”

If thousand:

Result += “ “ if result else “”

Result += three(thousand) + “ Thousand”

If remainder:

Result += “ “ if result else “”

Result += three(remainder)

Return result

Print(number\_to\_words(45))

Print(number\_to\_words(12345))

#2,

Def trailing\_zeros(n):

Count = 0

I = 5

While n // I > 0:

Count += n // i

I \*= 5

Return count

Print(trailing\_zeros(10)) # Output: 2 (since 10! = 3,628,800)