volSphere1 r =

let fHalf = 4 \* pi

lHalf = r^3 / 3

in fHalf \* lHalf

volSphere2 r = 4 \* pi \* r^3 / 3

numPow1 x 0 = 1

numPow1 x y = x \* numPow1 x (y-1)

numPow2 x y

| y ==0 = 1

| otherwise = x \* numPow2 x (y-1)

mymax a 0 = a

mymax a b

|a > b = a

|b > a = b

|otherwise = a

maxlist [] = 0

maxlist (head:tail) = mymax head (maxlist tail)

wsum = zipWith (\x y -> 2\*x+y)

sumofModNum = [x | x <- [0..999], x `mod` 3 == 0 || x `mod` 5 == 0]

equal3X 'X' 'X' 'X' = True

equal3X x y z = False

equal3 'X' 'X' 'X' = True

equal3 'O' 'O' 'O' = True

equal3 x y z = False

checkWin (x1:x2:x3:[]) (y1:y2:y3:[]) (z1:z2:z3:[])

|(equal3X y1 z2 x3) || (equal3X y1 x2 z3) = True

|(equal3X x1 z2 y3) || (equal3X x1 y2 z3) = True

|(equal3X z1 x2 y3) || (equal3X z1 y2 x3) = True

|otherwise = False

checkWinXorO (x1:x2:x3:[]) (y1:y2:y3:[]) (z1:z2:z3:[])

|(equal3 y1 z2 x3) || (equal3 y1 x2 z3) = y1:[]

|(equal3 x1 z2 y3) || (equal3 x1 y2 z3) = x1:[]

|(equal3 z1 x2 y3) || (equal3 z1 y2 x3) = z1:[]

|otherwise = "False"

checkMove3 \_ [] = 0

checkMove3 a (x:xs)

| a == x = 1 + checkMove3 a xs

|otherwise = 0 + checkMove3 a xs

checkBoard3 a b c

| ((checkMove3 'X' a + checkMove3 'X' b + checkMove3 'X' c) == 3) && ((checkMove3 'O' a + checkMove3 'O' b + checkMove3 'O' c) == 3) = True

|otherwise = False

wonInThreeByX x1 y1 z1

| checkBoard3 x1 y1 z1 = checkWin x1 y1 z1

| otherwise = False

wonInThree x1 y1 z1

| checkBoard3 x1 y1 z1 = checkWinXorO x1 y1 z1

| otherwise = "Lack of exactly 3 moves from either player X or O"

gameIsOver (x1:x2:x3:[]) (y1:y2:y3:[]) (z1:z2:z3:[])

| ((equal3 y1 z2 x3) || (equal3 y1 x2 z3)) = y1:[]

| ((equal3 x1 z2 y3) || (equal3 x1 y2 z3)) = x1:[]

| ((equal3 z1 x2 y3) || (equal3 z1 y2 x3)) = z1:[]

| otherwise = "False"