**Part 1 Pseudocode**

HIGHEST-PERFECT-SQUARE(n)

i <- 0

parameter <- 4 \* n

new <- []

for i <- 0 to n

i <- i + 1

perfectsquare <- i \* i

if perfectsquare <= parameter

append perfectsquare to new

return max[new]

**Part 3 Pseudocode**

CREATING-MATRIX(r, c, N)

matrixN <- []

for x <- 0 to r

row <- []

for y <- 0 to c

append N to row

append row to matrixN

return matrixN

ADDITION(r, c, N)

Bmatrix <- CREATING-MATRIX(r, c, N)

Cmatrix <- CREATING-MATRIX(r, c, N)

A <- [[0] \* r for i <- 0 to c]

for row <- 0 to length of Bmatrix

for column <- 0 to length of Cmatrix[row]

A[row][column] <- Bmatrix[row][column] + Cmatrix[row][column]

return A

MULTIPLICATION(r, c, N)

Bmatrix <- CREATING-MATRIX(r, c, N)

Cmatrix <- CREATING-MATRIX(r, c, N)

A <- [[0] \* r for i <- 0 to c]

for row <- 0 to length of Bmatrix

for column <- 0 to length of Cmatrix[row]

for row2 <- 0 to length of Bmatrix[0]

A[row][column] <- A[row][column] + (Bmatrix[row][row2] \* Cmatrix[row2][column])

return A

MULTIPLICATION-SC(r, c, N)

Bmatrix <- CREATING-MATRIX(r, c, N)

Cmatrix <- CREATING-MATRIX(r, c, N)

add = ADDITION(r, c, N)

A <- [[0] \* r for i <- 0 to c]

for row <- 0 to length of Bmatrix

for column <- 0 to length of Cmatrix[row]

A[row][column] <- 2 \* add[row][column]

return A

RESULT(r, c, N)

Bmatrix <- CREATING-MATRIX(r, c, N)

Cmatrix <- CREATING-MATRIX(r, c, N)

multi <- MULTIPLICATION(r, c, N)

multi\_Sc <- MULTIPLICATION\_SC(r, c, N)

A <- [[0] \* r for i <- 0 to c]

for row <- 0 to length of Bmatrix

for column <- 0 to length of Cmatrix[row]

A[row][column] <- multiI[row][column] – multi\_Sc[row][column]

return A