

Assignment (1) : Photo Editor

Names	ID
Osama Refaat Sayed Ali	20221015
Omar Ahmed Mohamed Saleh	20220220

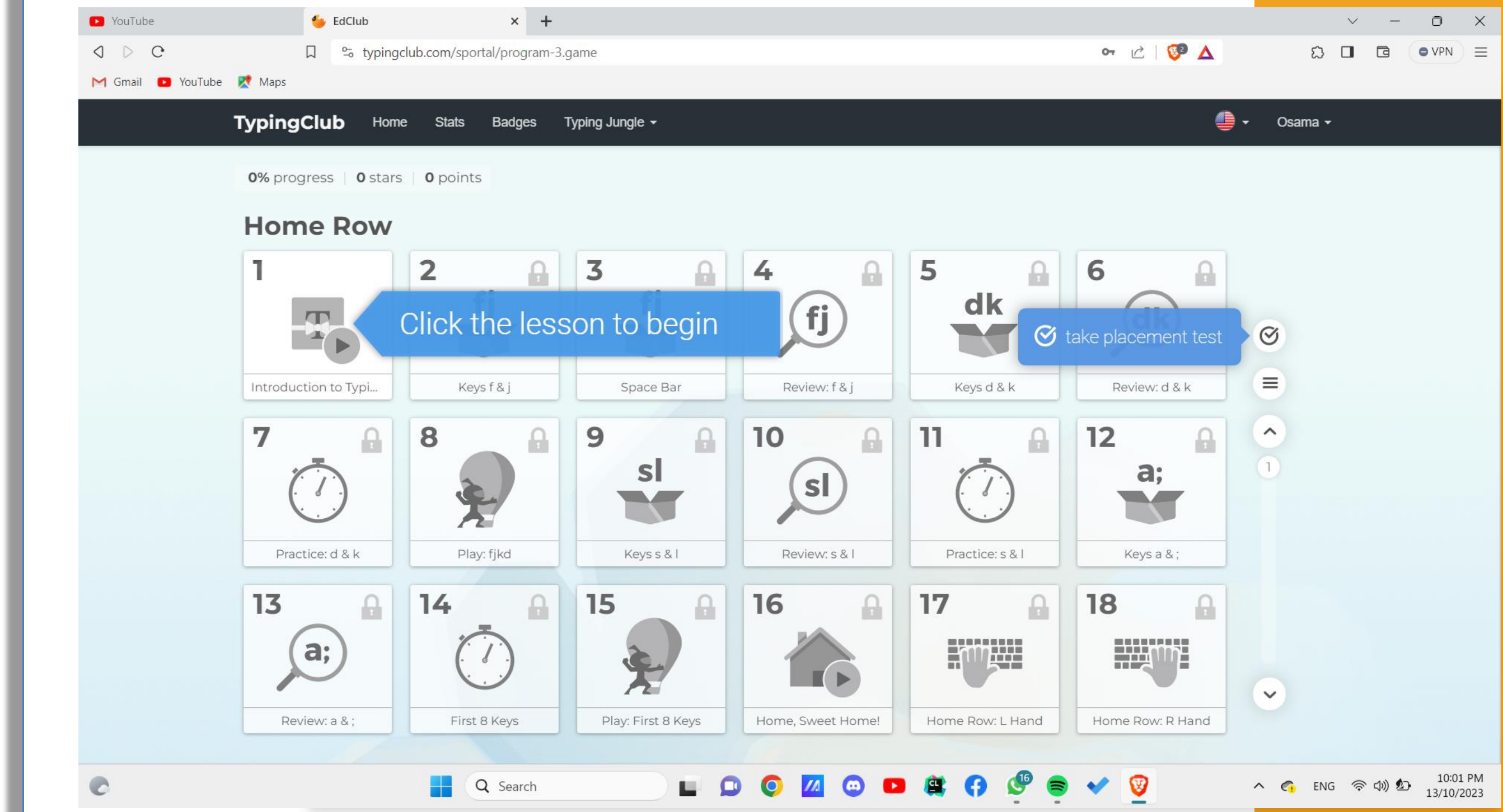




Typing Club

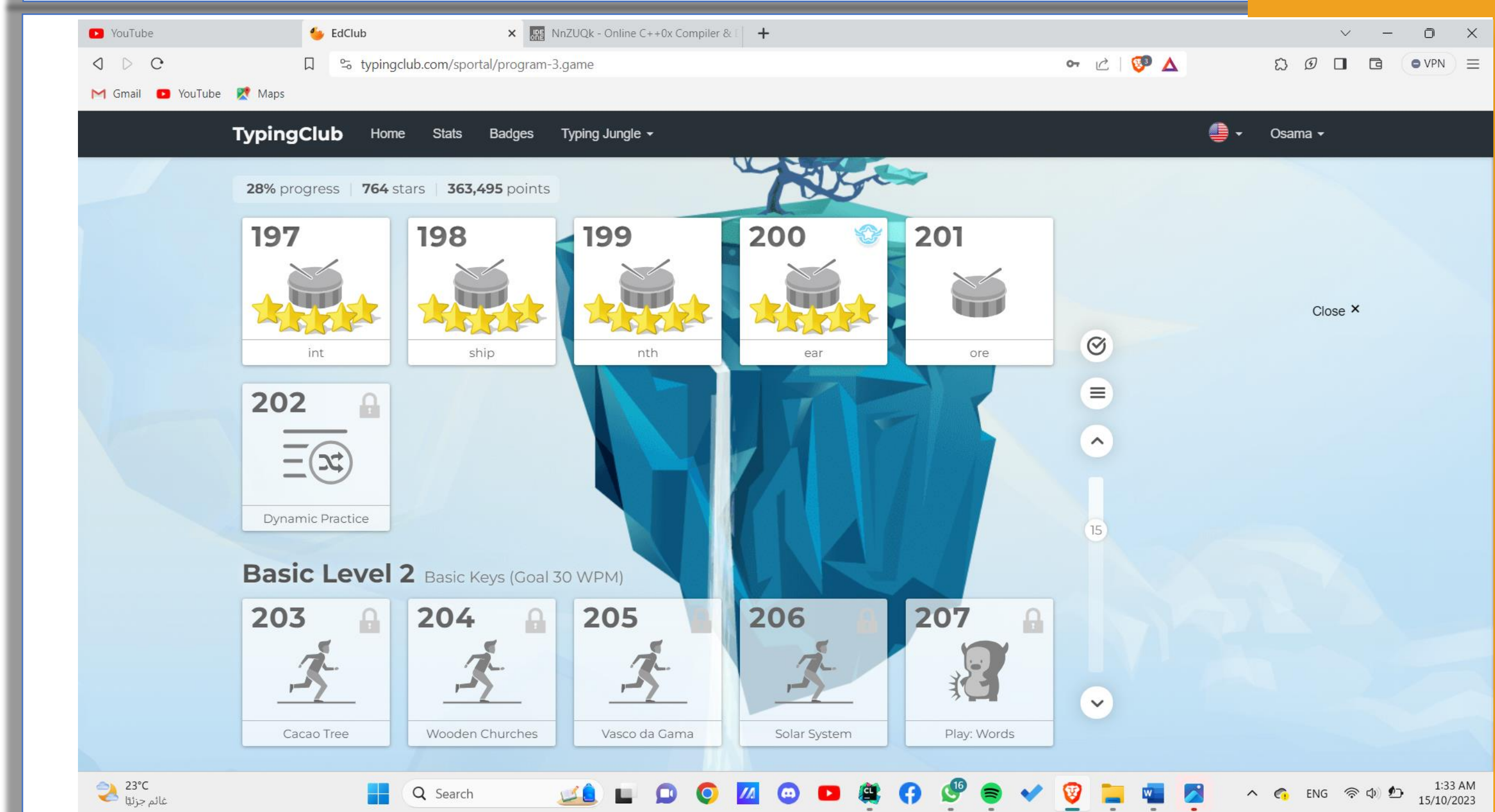


Typing club account on
13/10/2023 level zero

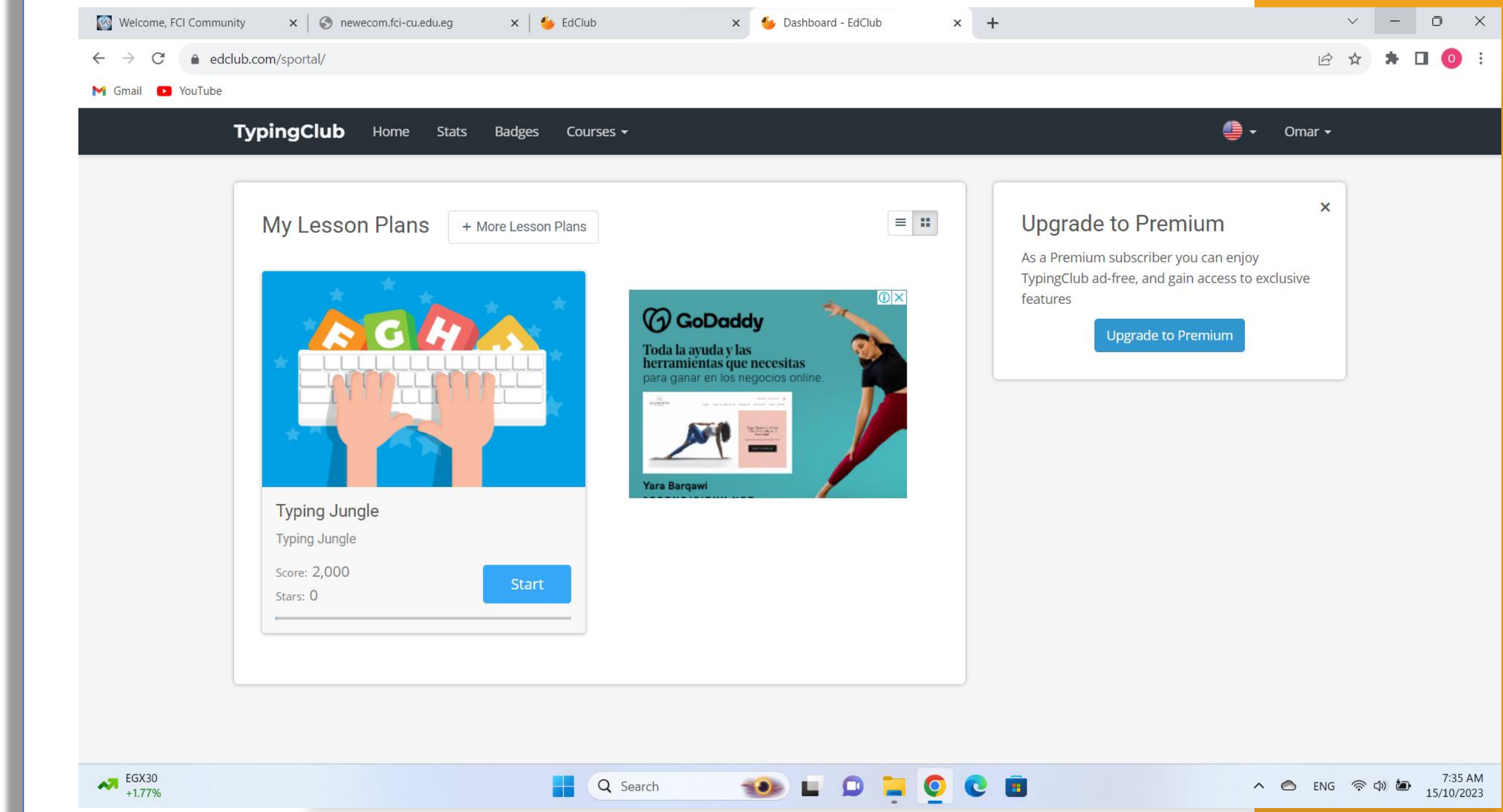


Osama Refaat
20221015
: Typing Club
Progress

Typing club account on
15/10/2023 level 200

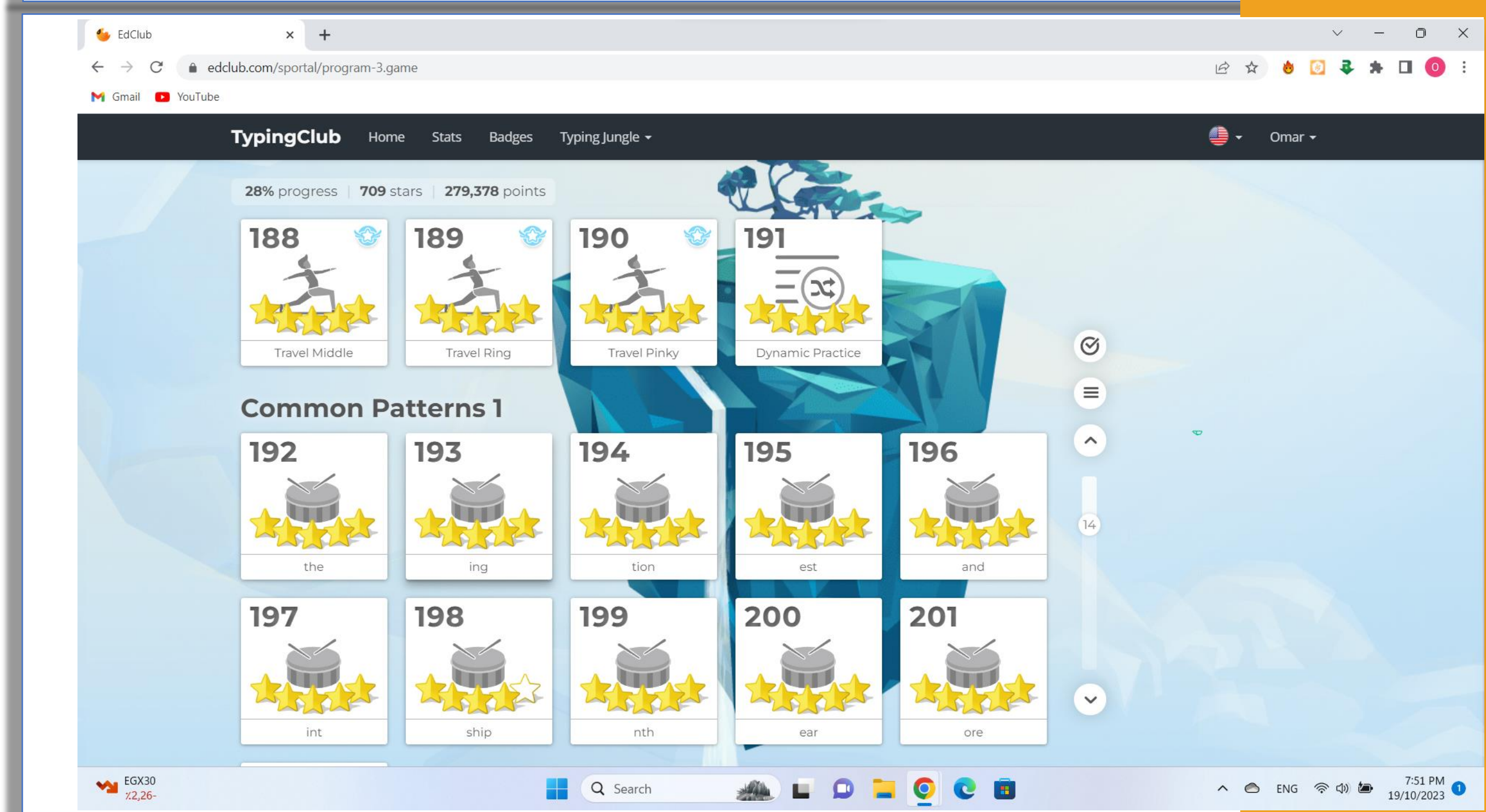


Typing club account on
15/10/2023 level zero



Omar Ahmed
20220220
: Typing Club
Progress

Typing club account on
19/10/2023 level 200





git



Github

Github : Workflow , commits

UsamaRefaat / photoeditor

Q Type to search

>

+ ▾

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Commits

main ▾

Commits on Oct 16, 2023

Merge pull request #2 from omaarsaaleh/patch-1 ...
UsamaRefaat committed 2 days ago

Update bonus.cpp ...
omaarsaaleh committed 2 days ago

Merge pull request #1 from omaarsaaleh/main ...
UsamaRefaat committed 2 days ago

Update README.md
omaarsaaleh committed 2 days ago

submitting my project cpp files
osama committed 2 days ago

Initial commit
UsamaRefaat committed 2 days ago

Verified

e7772b5

<>

Verified

168ac28

<>

Verified

0b2fd74

<>

Verified

75f9288

<>

73fd317

<>

Verified

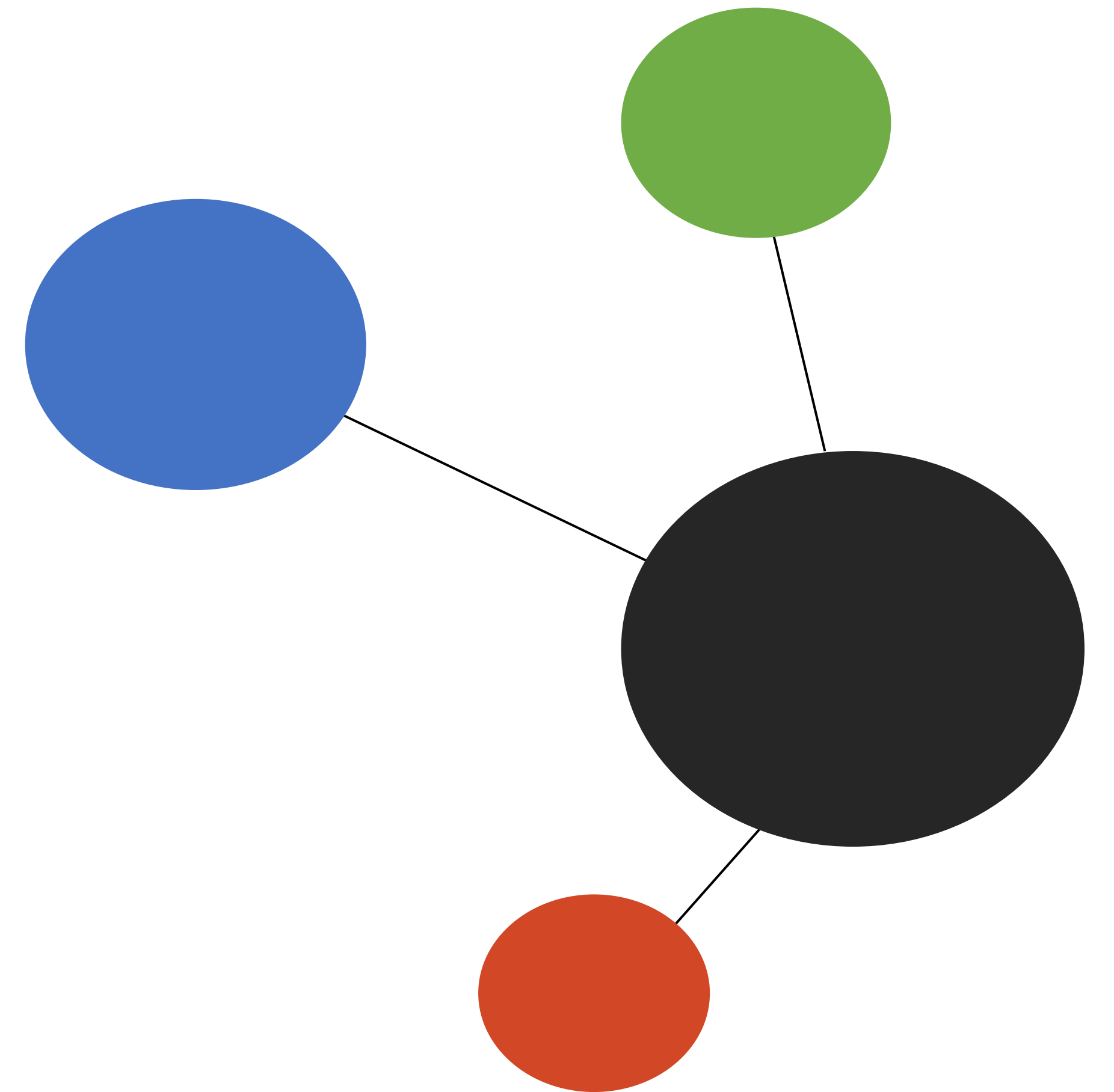
8b5a69b

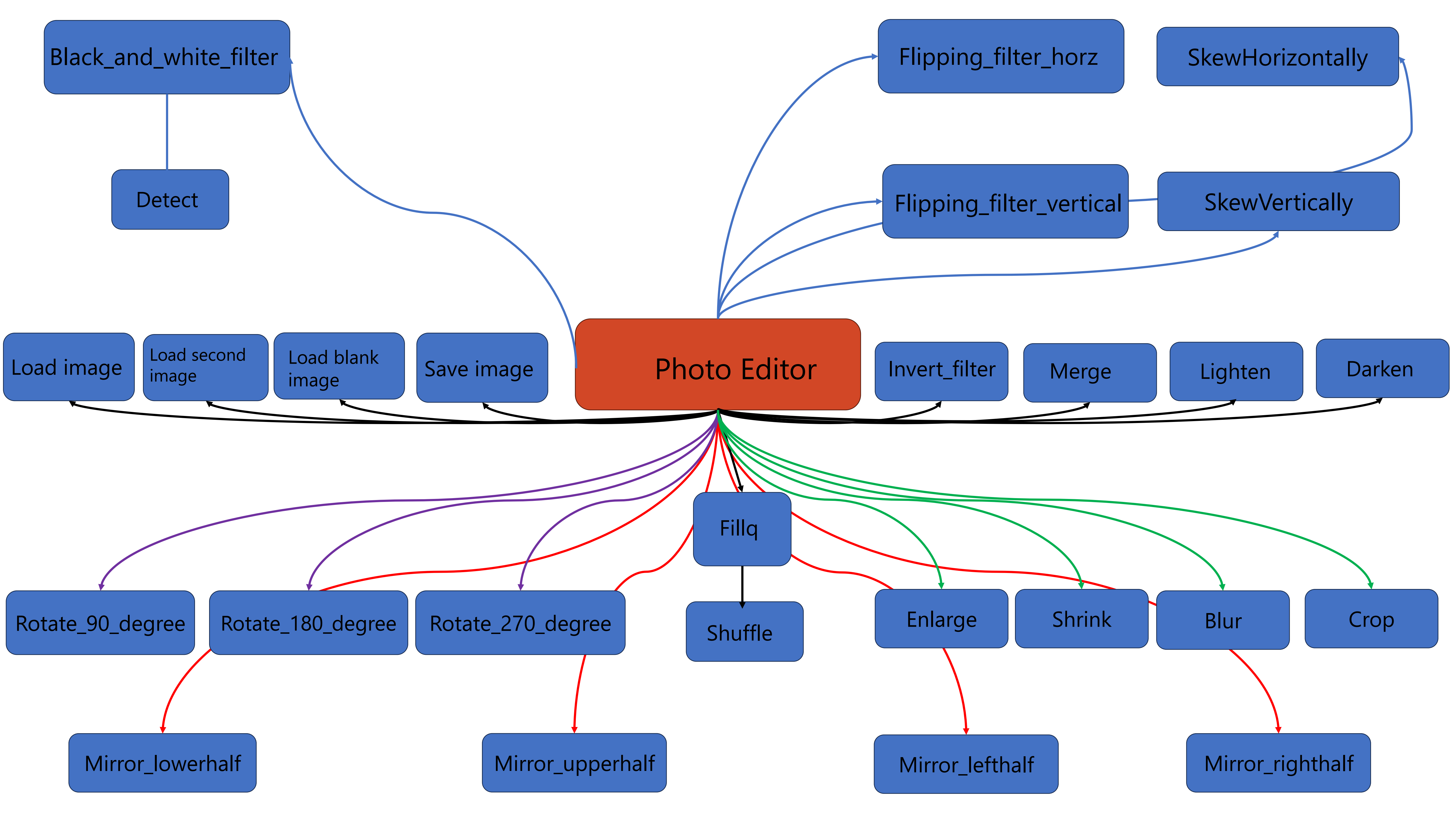
<>

Newer

Older

Function Decomposition







pseudocode

<p><i>function Black_and_white_filter:</i></p> <p>Initialize average to 0</p> <p>If not crop_shrink_skew: Calculate Average</p> <p>Else Average = 128</p> <p>For each Pixel If image[i][j] >= average: Set image[i][j] to 255 Else: Set image[i][j] to 0</p>	<p><i>function Invert_filter:</i></p> <p>For each Pixel Set image[i][j] to 255 - image[i][j]</p>	<p><i>function merge:</i></p> <p>For each Pixel Set image[i][j] to the average of image[i][j] and second_image[i][j]</p>
	<p><i>function flipping_filter_horz:</i></p> <p>For each i from 0 to 255: For each j from 0 to SIZE/2-1: Swap image[i][j] with image[i][SIZE-1-j]</p>	<p><i>function mirror_lowerhalf:</i></p> <p>For each i from 0 to SIZE/2-1: For each j from 0 to 255: Set image[i][j] to image[SIZE-1-i][j]</p>
	<p><i>function flipping_filter_vertical:</i></p> <p>For each i from 0 to SIZE/2-1: For each j from 0 to 255: Swap image[i][j] with image[SIZE-1-i][j]</p>	<p><i>function mirror_upperhalf:</i></p> <p>For each i from SIZE/2 to 255: For each j from 0 to 255: Set image[i][j] to image[SIZE-1-i][j]</p>
<p><i>function mirror_upperhalf:</i></p> <p>For each i from SIZE/2 to 255: For each j from 0 to 255: Set image[i][j] to image[SIZE-1-i][j]</p>		
<p><i>function mirror_LeftHalf:</i></p> <p>For each i from 0 to 255 For each j from SIZE/2 to 255: Set image[i][j] to image[i][SIZE-1-j]</p>	<p><i>function rotate_90_degree:</i></p> <p>For each i from 0 to 255: For each j from i+1 to 255: Swap image[i][j] with image[j][i]</p> <p>For each i from 0 to SIZE/2-1: For each j from 0 to 255: Swap image[i][j] with image[SIZE-1-i][j]</p>	<p><i>function rotate_180_degree:</i></p> <p>For each i from 0 to SIZE/2-1: For each j from 0 to 255: Swap image[i][j] with image[SIZE-1-i][SIZE-1-j]</p>

<p><i>function rotate_270_degree:</i></p> <p>For each i from 0 to 255: For each j from i+1 to 255: Swap image[i][j] with image[j][i]</p> <p>For each i from 0 to 255: For each j from 0 to SIZE/2-1: Swap image[i][j] with image[SIZE-1-i][j]</p>	<p><i>function lighten:</i></p> <p>For each i from 0 to 255: For each j from 0 to 255: Add (255 - image[i][j]) / 2 to image[i][j]</p> <p><i>function darken:</i></p> <p>For each i from 0 to 255: For each j from 0 to 255: Subtract image[i][j] / 2 from image[i][j]</p>	<p><i>function detect:</i></p> <p>Call Black_and_white_filter function</p> <p>For each i from 1 to 254: For each j from 1 to 254: If image[i][j] is 0 and at least one of its neighboring pixels is non-zero: Set blank_image[i][j] to 0 Else: Set blank_image[i][j] to 255</p> <p>Swap image and blank_image</p>
<p><i>function crop:</i></p> <p>output "Please enter x, y, l, w:" input x, y, l, w from the user</p> <p>Call load_blankimage function</p> <p>Calculate length_end as x + l Calculate width_end as y + w</p> <p>For each i from x to length_end-1: For each j from y to width_end-1: Set blank_image[i][j] to image[i][j]</p> <p>Swap image and blank_image</p>	<p><i>function shrink:</i></p> <p>Display "Shrink to (1/2), (1/3) or (1/4)?" Display "Enter '2' for (1/2), '3' for (1/3), '4' for (1/4)" Read n from the user</p> <p>Call load_blankimage function</p> <p>Set size to 256 divided by n Initialize x to 0 and y to 0</p> <p>For each Pixel Set blank_image[i][j] to image[x][y] Increment y by n</p> <p>Increment x by n Swap blank_image and image</p>	<p><i>function fillq(s, e, q):</i></p> <p>Calculate sizei as s + 128 Calculate sizej as e + 128</p> <p>Initialize x, y, and yp based on the value of q</p> <p>For each i from s to sizei-1: For each j from e to sizej-1: Set blank_image[i][j] to image[x][y] Increment y by 1</p> <p>Increment x by 1 Set y to yp</p>

function shuffle:

Display "Enter the order of quarters you would like in your new image"

Read a, b, c, d from the user

Create a map mp and increment its count for each quarter value

For each i from 1 to 4:

 If mp[i] is not 1:

 Display "Invalid Input"

 Return

Call load_blankimage function

Call fillq with (0, 0, a) to fill the first quarter

Call fillq with (0, 128, b) to fill the second quarter

Call fillq with (128, 0, c) to fill the third quarter

Call fillq with (128, 128, d) to fill the fourth quarter

Swap blank_image and image

function enlarge:

Display "Please Choose which quarter to Enlarge '1','2','3','4'"

Read n from the user

Call load_blankimage function

If n is 1:

 Set x to 0, y to 0, and py to 0

Else if n is 2:

 Set x to 0, y to 128, and py to 128

Else if n is 3:

 Set x to 128, y to 0, and py to 0

Else if n is 4:

 Set x to 128, y to 128, and py to 128

Else:

 Display "Invalid Input"

 Return

For each Pixel

 Set blank_image[i][j] to image[x][y]

 If j is even, increment y by 1

If i is even, increment x by 1

Set y to py

Swap blank_image and image

function skewHorizontally:

Load blankimage

Display "Please enter an angle in degrees between 0 and 45"
Read angle from the user

If angle is less than 0 or greater than 45:
 Display "Invalid Input"
 Return

Calculate tanrad as the tangent of angle converted to radians

Initialize current_side to SIZE
Initialize paralloSide to $SIZE - SIZE * \tan\text{rad}$
Initialize skip factor as 1 if angle is 45.0, else as $1 / (1 - \tan\text{rad})$

Initialize y to 0

For each i from 0 to SIZE-1:
 For each j from current_side * tanrad to current_side * tanrad + paralloSide:
 Set blank_image[i][j] to image[i][y]
 Increment y by skip

Swap blank_image and image

function skewVertically:

Load blankimage

Display "Please enter an angle in degrees between 0 and 45"
Read angle from the user

If angle is less than 0 or greater than 45:
 Display "Invalid Input"
 Return

Calculate tanrad as the tangent of angle converted to radians
Initialize current_side to SIZE
Initialize y to 0
Initialize rightside as $SIZE * \tan\text{rad}$
Initialize skip as 1 if angle is 45.0, else as $1 / (1 - \tan\text{rad})$

For each col from 0 to SIZE-1:
 For each row from current_side * tanrad to SIZE - rightside + current_side * tanrad:
 Set blank_image[row][col] to image[y][col]
 Increment y by skip

Swap blank_image and image

function blur:

Call load_blankimage function

For each i from 0 to SIZE-1:

For each j from 0 to SIZE-1:

Initialize sum to 0

Initialize count to 0

For each x from -2 to 2:

For each y from -2 to 2:

Calculate x1 as i + x

Calculate y1 as j + y

If x1 is within 0 and SIZE-1 and y1 is within 0 and SIZE-1:

Add image[x1][y1] to sum

Increment count by 1

Set blank_image[i][j] to the integer division of sum by count

Swap blank_image and image