

**Cairo university**

**Faculty of computer and artificial intelligence**

**Cs112 – Structured programming**

**Second semester 2021-2022**

**Group: B S16**

**Assignment #3 Report**

**TA: Eng-Nesma / Eng-Yousra**

**Adel Magdy Abd El-Hay (20210190)**

[**adool.ashoor357@gmail.com**](mailto:adool.ashoor357@gmail.com)

**Phone number: 01099687903**

**Asmaa Saleh Farghaly** **(20211014)**

[**asmaasalehnoon@gmail.com**](mailto:asmaasalehnoon@gmail.com)

**Phone number: 01554884939**

**Roaa Talat Mohamed ( 20210138)**

[**roaatalat211@gmail.com**](mailto:roaatalat211@gmail.com)

**Phone number: 01121295650**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Filters** | **Filters-bonus** |
| **Adel Magdy Abd El-Hay** | **20210190** | **1,4,7,a** | **1,4,7,a** |
| **Asmaa Saleh Farghaly** | **20211014** | **2,5,8,b** | **2,5,8,b** |
| **Roaa Talat Mohamed** | **20210138** | **3,6,9,c** | **3,6,9,c** |

**Algorithm**

1. **Ask the user to choose the number/ litter of the filter**
2. **If the number is 1(Black & White Filter)**

* **Call the functions(loadImage() convertImageToBlackAndWhiteImage() and saveImage())**
* **Make a nested loop that passes over each pixel in the image and take the average of all pixels**
* **Again, make a nested loop that passes over each pixel in the image to convert all pixels to black and white with condition**
* **If the value of pixel greater than average convert it to white**
* **If not convert it to black**
* **Then save the image**

1. **if the number is 2(Invert Filter)**

* **Call the functions (loadImage()), (Invert()) and saveImage(()**
* **Use a nested loop , once loop to represent pixel in raws ,and second loop to represent pixel in columns**
* **Subtract from 255 each pixel value to represent it with the inverse.**
* **Save the image .**

1. **if the number is 3( Merge Filter)**

* **call the functions ( loadImage() ) , (loadSecondImage()) ,(** **mergeImage() )and (saveImage() )**
* **ask the user to enter the name of the first and second image**
* **make a nested loop that passes over each pixel**

**in the first and second image**

* **add each pixel in the first image to the second image then divide them by 2**
* **save the image**

1. **if the number is 4(Flip Image)**

* **Call the functions(loadImage() , flipImage() and saveImage())**
* **Make a nested loop that passes over each pixel in the image and reverse elements of every column**

1. **if the number is 5(Rotate Image)**

* **Call the functions (loadImage()),(rotateImage()) and (saveRotate()).**
* **Ask the user which degree of rotate he want .(90 -180 – 270 )**
* **If user chose (90) , call the function (rotate -90-degree)**
* **Use nested loop to represent raws and columns .**
* **Turn each raw into a column and subtract the column from the size and put the result in the new raw.**
* **If user choose (180) , call the function (rotate -180-degree)**
* **Use nested loop to represent rasa and columns .**
* **Subtract the columns and raws from size .**
* **If user chose (270) , call the function (rotate -270-degree)**
* **Use nested loop to represent raws and columns .**
* **Turn each column into rows and subtract the row from the size and put the result in new column.**
* **Save the image .**

1. **if the number is 6(Darken and Lighten Image)**

* **call the functions ( loadImage() ) , and (saveImage() )**
* **ask the user to enter the name of the image**
* **ask the user to enter (d) for darken and (l) for lighten**
* **if he enter (d) call the function (darkenImage())**
* **make a nested loop that passes over each pixel**

**then** **multiply each pixel by 0.5**

* **if he enter (l) call the function ( lightenImage() )**
* **make a nested loop that passes over each pixel**

**then** **multiply each pixel by 1.5 and check if it is bigger than 255 let it it be 255**

* **save the image**

1. **if the number is 7(Detect Image Edges)**

* **load the image**
* **call function DetectImageEdges()**
* **2 nested loops(loop for rows and for columns) to loop in each pixel**
* **if result of subtract two adjecant pixels greater than or equal to 20 (convert the first pixel to black)**
* **else (convert the first pixel to white)**
* **save the image**

1. **if the number is 8(Enlarge Image)**

**10.if the number is 9(Shrink Image)**

* **call the functions (loadImage() ) , and (saveImage() )**
* **ask the user to enter the name of the image**
* **ask the user to enter (½) to shrink the image to half, (1/3) to shrink the image a third and (¼) to shrink the image a quarter**
* **if he enter (½) call the function (shrinkAhalfImage())**
* **make a nested loop that passes over each pixel in the rotate image be white (255)**
* **make a nested loop that passes over each pixel , take the average of 4pixels (square) from image then put it in the rotate image**
* **if he enter (1/3) call the function (shrinkAthirdImage())**
* **make a nested loop that passes over each pixel in the rotate image be white (255)**
* **make a nested loop that passes over each pixel, take the average of 9pixels (square) from image then put it in the rotate image**
* **if he enter ( ¼ ) call the function (shrinkQuarterImage())**
* **make a nested loop that passes over each pixel in the rotate image be white (255)**
* **make a nested loop that passes over each pixel, take the average of 16pixels (square) from image then put it in the rotate image**
* **save the rotate image**

**11.if the litter is a(Mirror 1/2 Image)**

* **load image**
* **call function mirrorImage()**
* **give the user choice to choose which side to mirror**
* **if the choice equal to ‘l’ call mirrorLeftRight()**

1. **we make 2 nested loop (rows and columns)**
2. **the loop in columns to (SIZE / 2) because we mirror half of image**
3. **mirror each pixel 🡪 image[row][255-col] = image[row][col]**
4. **save image**

* **if the choice equal to ‘r’ call mirrorRightLeft()**

1. **make 2 nested loop (rows and columns)**
2. **the loop in columns to (SIZE / 2) because we mirror half of image but we inverse the counter to start from the end**
3. **mirror each pixel 🡪 image[row][255-col] = image[row][col]**
4. **save image**

* **if the choice equal to ‘u’ call mirrorUpDown()**

1. **make 2 nested loop (rows and columns)**
2. **the loop in rows to (SIZE / 2) because we mirror half of image**
3. **mirror each pixel 🡪 image[255-row][ col] = image[row][col]**
4. **save image**

* **if the choice equal to ‘d’ call mirrorDownUp ()**

1. **make 2 nested loop (rows and columns)**
2. **the loop in rows to (SIZE / 2) because we mirror half of image but we inverse the counter to start from the end**
3. **mirror each pixel 🡪 image[255-row][ col] = image[row][col]**
4. **save image**

* **else print (invalid side)**

**12.if the litter is b(Shuffle Image)**

**13.if the litter is c(Blur Image)**

* **call the functions ( loadImage() ) , (blurImage()) and (saveImage() )**
* **ask the user to enter the name of the image**
* **make a nested loop that passes over each pixel**
* **then make another nested loop in the first one that take a square of pixels (7rows, 7colunms)**
* **calculate the sum of these pixels**
* **after finishing the second nested loop let the average of these pixels in the middle of the square**
* **save the image**

**14.if the number is 0 print (See you later).**