

Program 2: Create a collage (30 marks)

- a) Choose the picture you wish to use in your collage.

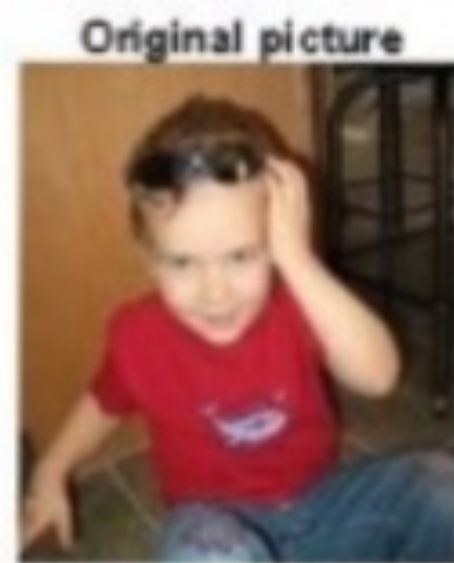
You can (a) download and use a personal picture file (.jpg or .png) in your collage, (b) use the picture you created in the previous program in your collage, OR (c) you can choose a picture from those provided in the 'L10media.zip' in Blackboard. When tested, your program should work properly on **any picture** (minimum 100 pixels by 100 pixels).
NOTE: Large personal pictures should be cropped prior to use.

- c) Implement **FOUR** helper functions as outlined next. **Test each one as you create it!** Examples are provided to show you how to do this. Also, ensure each function has its own function purpose in comments.

- `inverse(file_name):`

Use the file name passed to create a picture. Create an inverse rectangle along the right side of the picture keeping the rest of the original picture intact. The rectangle should be a quarter of the width and the entire height. To convert a pixel to its inverse, set each colour component (red, green, and blue) to the maximum value possible (255) minus its current value. Label the picture 'Inverse' at position (5, 5) in the colour gold (255, 215, 0) (as shown below) and return the new picture.

Example
using
"will.jpg":



```
>>> inverse("will.jpg").show()
```



- `tile(file_name):`

Use the file name passed to create a picture. Create a tiled row along the bottom of the picture keeping the rest of the original picture intact. A tiled row can be done by (a) resizing the original picture to 1/16th of its size (1/4 width and 1/4 height), and then (b) pasting the resized image four times along the bottom of the original picture.

In addition, the function must **ask the user** to decide whether to rotate the resized image 180 degrees or not. No input error checking is required but the input should be case insensitive. If the user indicates "yes", perform the rotation **before** pasting the resized image in (b). Label the picture either 'Tile' or 'Tile and rotate' at position (5, 5) in the colour gold (255, 215, 0) (as shown below) and return the new picture.

```
>>> tile("will.jpg").show()
Rotate tile (yes or no)? n0
```



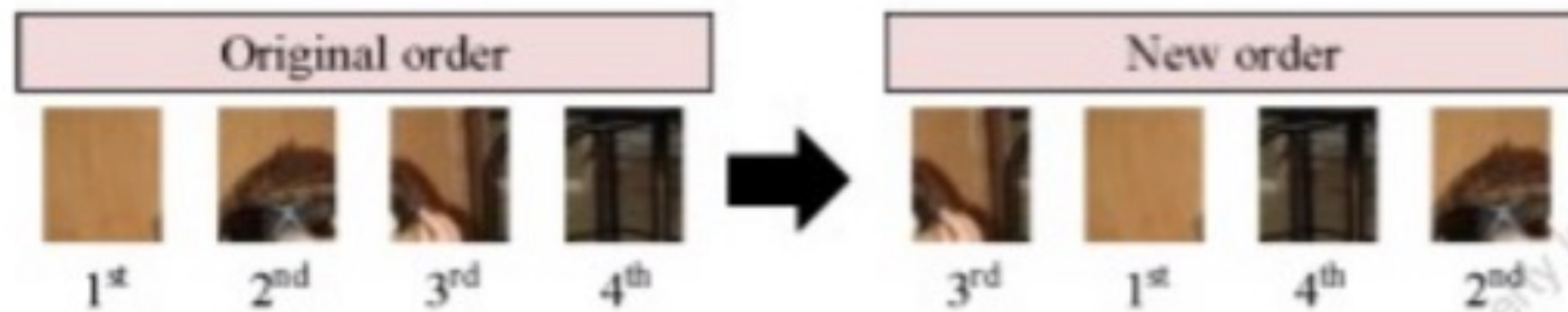
```
>>> tile("will.jpg").show()
Rotate tile (yes or no)? yEs
```



Examples
using
"will.jpg":

- `puzzle(file_name)`

Use the file name passed to create a picture. Create a puzzle along the top of the picture keeping the rest of the original picture intact. Do this by (a) cropping the top part of the picture into four images that are each a quarter of the width and a quarter of the height, and then (b) pasting each image back into the original picture in a new position along the top part of the picture. The images should be ordered as follows:



Label the picture 'Puzzle' at position (5, 5) in the colour gold (255, 215, 0) (as shown below) and return the new picture.

```
>>> puzzle("will.jpg").show()
```

Example
using
"will.jpg":



- `greyscale(file_name):`

Use the file name passed to create a picture. Create a greyscale rectangle along the left side of the picture keeping the rest of the original picture intact. The rectangle should be a quarter of the width and the entire height. To convert a pixel to greyscale, set its colour to the average of its three colour components (red, green, and blue). Label the picture 'Greyscale' at position (5, 5) in the colour gold (255, 215, 0) (as shown below) and return the new picture.

```
>>> greyscale("will.jpg").show()
```

Example
using
"will.jpg":



- d) Implement the main function in your program: `make_collage`
 This will create a 4 picture collage (2 pictures across and 2 pictures below).

When `make_collage(file_name)` is called, your program should:

- (a) Call all necessary helper functions to perform the required transformations
(explained earlier).

- (b) Create an empty canvas large enough to accommodate four pictures (2 x 2)

- (c) Paste the transformed pictures into the empty canvas to create the collage using the outline as shown.

Quadrant 1: inverse	Quadrant 2: tile
Quadrant 3: puzzle	Quadrant 4: greyscale

- (d) Show the **final** collage only (do not show intermediate collage results)

- (e) Save the collage as `'collage.png'` *(excluding the quotes)*.

Sample collages using the possible transformation functions are shown next. Quadrant 2 varies depending upon the user's choice.



- e) Save your program as `"L10Q2initials.py"` replacing *initials* with your actual initials.

- f) **Submit** your file electronically for marking (unless otherwise instructed).