Data

- Image - 2D array, size to be determined
- Imageptr - pointer to 2D array
- Filename - string array, to save name of file
- Choice1 - to save user’s menu choice
- fileptr - file pointer

Functions

Main function

- Display menu
    - selection 1
        - Declare file pointer
        - Run load image function
        - save result in file name variable
        - Open file for re
    - selection 2
        - Run display image function
    - selection 3
        - User chooses: crop, dim, brighten, or rotate
        - Run appropriate function
        - Run display image function
        - Get file name
        - Save new file with that name (run create new file function)
    - selection 4: exit program
        - Close file
        - End runtime
- Repeat until selection 4 is chosen

Load a new image

input parameters: file pointer, 2D array pointer
Returned output: bool (1=success, 0=fail)

- Get file name
- Return file name
- Open file for reading
    - If unable to open
        - Display error message
        - return 0
- Write data to 2D array pointer

```
    -   Close file
    -   Return 1

Display current image
        input parameters: 2D array pointer
        Returned output: nothing (display only)

    -   For each row
            -   For each column index
                    -   If stored value is 0, display ‘ ‘
                    -   If stored value is 1, display ‘.’
                    -   If stored value is 2, display ‘o’
                    -   If stored value is 3, display ‘O’
                    -   If stored value is 4, display ‘0’
                    -   If stored value is 5, display nothing
            -   Display new line (\n)

Crop
        input parameters: 2D array pointer
        Returned output: updating array (via pointers), returns nothing

    -   Get user choice: crop the sides, yes or no
            -   Get user choice: left, right, or both
            -   Get how many columns to crop off (#)
            -   Right - for each row:
                    -   for each column index
                            -   ascending increment
                            -   set the value equal to the value in the index # places higher
                            -   for the last # indexes, store the value ‘5‘
            -   Left - for each row:
                    -   for each column index
                            -   descending increment
                            -   set the value equal to the value stored in the index # places lower
                            -   for the first # indexes, store the value ‘5‘
            -   Both
                    -   same code as ‘right’ section
                    -   same code as ‘left’ section
    -   Get user choice: crop the top/bottom, yes or no
            -   Get user choice: top, bottom, or both
            -   Get how many rows to crop off (#)
            -   Top - for each column
                    -   for each row index
                            -   ascending increment
                            -   set value equal to the value in the index # places higher
```

```
                - for the last # indexes, store the value ‘5‘
        - Bottom - for each column
            - for each row index
                - Descending increment
                - set value equal to the value in the index # places lower
                - for the first # indexes, store the value ‘5‘
        - Both
            - same code as ‘top’ section
            - same code as ‘bottom’ section
```

Dim

input parameters: 2D array pointer

Returned output: updating array (via pointers), returns nothing

```
    - Open file for writing
    - For every index (row and column) of the 2D array
        - If brightness value is less than 4
            - Add 1 to the brightness value
```

Brighten

input parameters: 2D array pointer

Returned output: updating array (via pointers), returns nothing

```
    - For every index (row and column) of the 2D array
        - If brightness value is greater than 0 and less than 5
            - subtract 1 from the brightness value
```

Rotate 90 degrees (clockwise)

input parameters: 2D array pointer

Returned output: updating array (via pointers), returns nothing

```
    - Declare a new 2D array (array 2)
        - The sizes of the columns and the rows should be inverse to the original
    - For each column in array 2
        - For each row index
            - Get new value location
                - col index 2 =  row index 1
                - row index 2 = max row 2 index - col index 1
            - Save value as the value that is stored in array 1 at the calculated indexes
```

Create new file

input parameters: file name, 2D array name

Returned output: updating file (via pointers), returns nothing

- Create new file with the name
- Enter data from input 2D array
- Close file