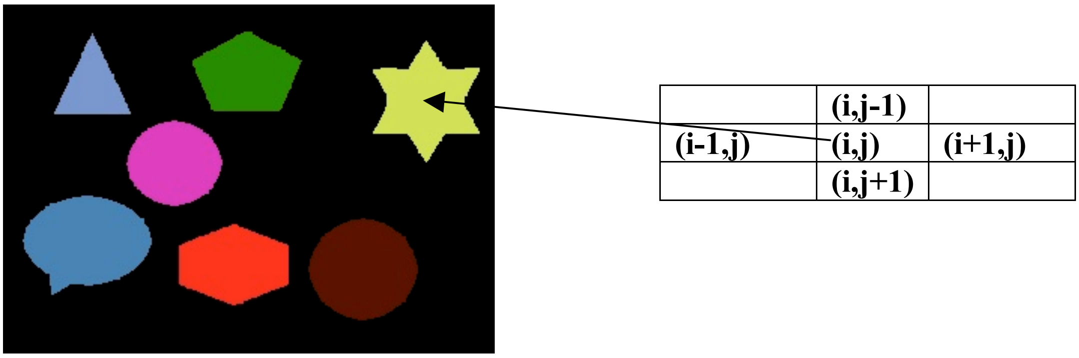
NYUAD Computer Systems Programming PROJECT 3

Project Deadline: May 11, 2017, 2:00pm, No late submission

In this project, you are going to **find out connected components** in the image. Your program will read a text file which contains floating format pixel (0 ~1) values of the image. We will denote the set of 2-D lattice points by S and individual lattice points by

Note:

1. Each connected component will be found based on a region-growing algorithm and some important concept provided below.
2. For-loop to find all possible connected component
3. You will define a class for image with definition of data member and function members
4. Neighbors definition:



	(i,j-1)	
(i-1,j)	**(i,j)**	**(i+1,j)**
	(i,j+1)	

In this project, we use 4-connected neighbors as indicator above. For each pixel position (i,j), we will have four directly connected neighbors.

5. Two steps involved in connected components detection algorithm:
   - For a given pixel (seed pixel), find out the connected component based on region grow. Region is grown from the seed pixel by adding in neighboring pixels that are similar, increasing the size of the region. The detailed process is described below:

Step 1	Given a seed point (i,j)
Step 2	Determine the neighbors coordinates
Step 3	Add the neighbors to potential candidates of connected components to seed point
Step 4	Compute the difference between candidate pixel and the mean of all of identified connected pixels
Step 5	Pick up the nearest candidate pixel as the newest identified connected component and update the seed point using the newest found point
Step 6	Repeat 1-5 above till termination criteria meets

Note: You can refer to http://www.mathworks.com/matlabcentral/fileexchange/19084-region-growing for a matlab implementation of region growing.

   - Indexing through the image in raster order and applying the first step at each pixel that does not yet belong to a connected region.

6. Output of your program should be number of the connected components.

```
//Function to read image from a file.
//The pixel position (i,j) corresponds the index (i*h + j) in an image array, h is the
//height of the image

double* getimage(string filename)
{
  //define a ifstream object
  ifstream myfile (filename);
  int w; // Image Width
  int h; // Image Height
  string line; // Each line from the image file
  getline (myfile,line); // Get Each Line
  w = atoi(line.c_str()); // Convert to the interger number
  getline (myfile,line);  // Get the second line
  h = atoi(line.c_str()); // Conver to the interger number
  int len = w*h;          // Allocate the len of memory for storing the image
  double *img = new double[len];
  int i =0;               // While-loop to read the file
  if (myfile.is_open())
  {
    while ( getline (myfile,line) )
    {
              img[i] = atof(line.c_str());
              i++;
    }
    myfile.close();
  }
  return img;             // return image array pointer
}
```