**חלק ראשון:**

**Preproccessing:**

**Input: filename1, filename2  
output: g1, g2**

1. *Try to load into g1 the saved data from file named filename1*
2. *If the number of vertices equals 0*
   1. *Detect the starts using image processing and save it into g1*
3. *Try to load into g2 the saved data from file named filename2*
4. *If the number of vertices equals 0*
   1. *Detect the starts using image processing and save it into g2*
5. *Return g1, g2*

***Brute force transformation search:***

1. *Best\_score = 0*
2. *take 3 points from g1 and create a triangle t1*
3. *take 3 points from g2 and create a triangle t2*
4. *local\_score = 0*
5. *If t1 is not similar to t2 with error smaller than* 
   1. *Look for different triangles*
6. *Create a transformation matrix using the 3 input points from g1 and the assumed output point from g2.*
7. *Match the transformed point to its closest neighbor within the distance threshold, and keep the distance****note:*** *If we can find a closer match we will take it instead*
8. *For each match add the local score (distance threshold – distance)*
9. *If local\_Score > best\_score*
   1. *Best\_score = local\_score*
   2. *Best\_matching = matching*
10. *for each 3 stars in g1*
    1. *For each 3 stars in g2*
       1. *Repeat phases 2 - 9*
11. *Return the best\_matching, transformed points*

***Algorithm:***

***Input: filename1, filename2***

***Output: best matching between both images***

1. *Preprocessing the images into graphs g1 and g2*
2. *Try to find the best transformation*
3. *Return the best match*