**שאלה 1:**

1. לפי נתון של הקואורדינאטות נחשב כמה אחוזים של כל אחד משלושת המקורות  *נדרש ליצירת צבע :*
2. *נתון נוכיח כי לפי נתון מתקיים: כלומר נקבל:*

*נציב את הנתונים שקיבלנו לתוך משווה ונקבל:*

נוציא שבר מהסוגריים נקבל:

נכפיל צד ימין של משווה ב-C ותוצאה סופית שנקבל:

מש"ל.

**שאלה 2:** התוכנה כתובה ב-MATLAB שמקבלת כקלט תמונה ומציגה אותה ישירות על המסך ב-FIGURE1 ואחרי 3 שניות נפתח חלון חדש FIGURE2 כך שמוצגת שם אותה תמונה אך רכיב אדום וירוק מוחלף.התמונה החדשה שהתקבלה ב- FIGURE2נשמרת כקובץ PNG.התוכנה יראה ככה:

% Read image.

image = imread ('liverpool.jpg');

% Show image in Figure 1

imshow(image);title('Original')

% Wait 3 seconds

pause(3);

% Now swap color channels.

image\_swap(:,:,1)= image(:,:,2);

image\_swap(:,:,2)= image(:,:,1);

image\_swap(:,:,3)= image(:,:,3);

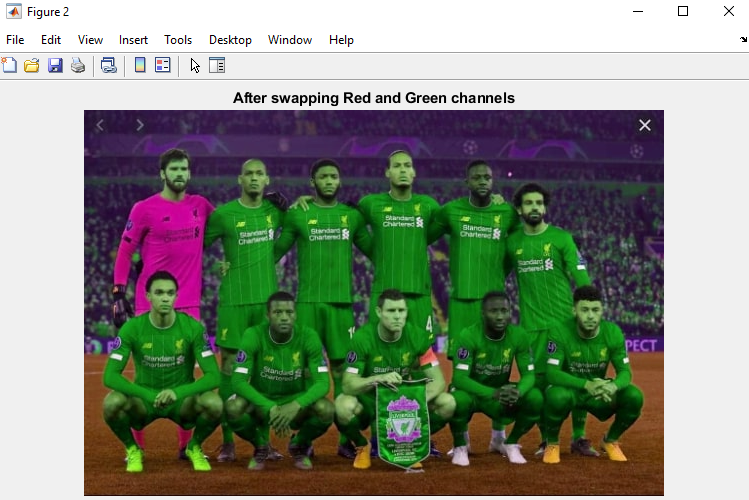
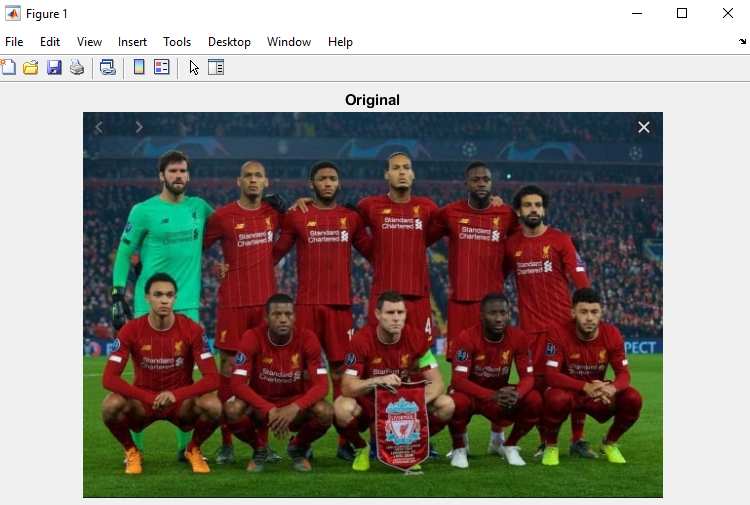
% Display swapped image in Figure 2.

figure,imshow(image\_swap);title('After swapping Red and Green channels')

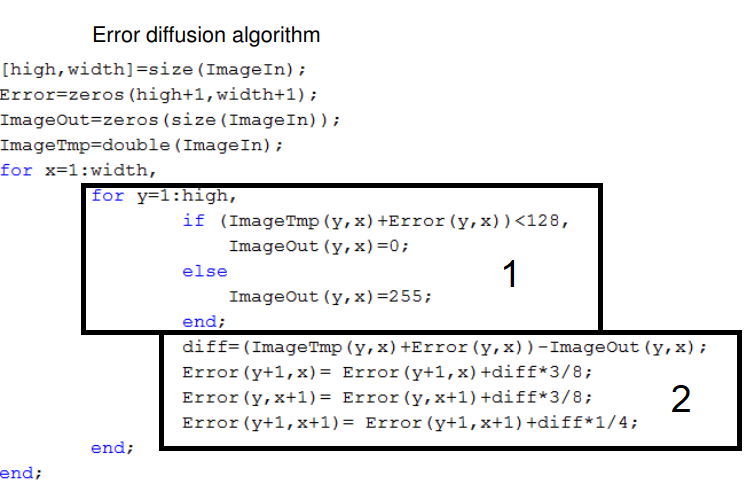
% Save swapped image.

imwrite(image\_swap,'OurPicture.png');

פלט שנקבל על המסך:



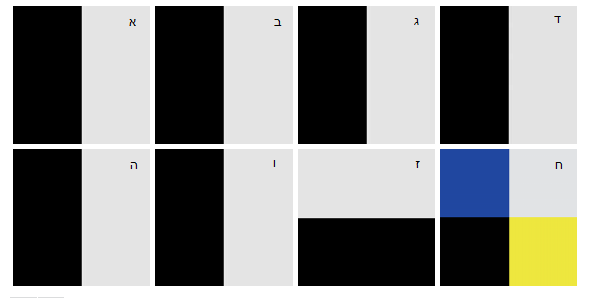
**שאלה3:** נדגים את אלגוריתם ERROR DIFFUSION :

אם נסתכל לקוד של אלגוריתם ERROR DIFFUSION,אז אפשר לראות שברגע שהפעלנו פעם אחד אלגוריתם אז לפי סימון 1 שיש בתמונה נקבל שכל פיקסל אחרי סיום של האלגוריתם אקבל ערך 255 שזה צבע לבן או 0 שזה צבע שחור.ולכן עכשיו נרצה להפעיל אלגוריתם עוד פעם ולראות האם יהיה שינוי כלשהו אחרי שיצרנו תמונה בינארית שמכילה רק ערכים 255 או 0.ולכן נפריד לשני מקרים,מקרה ראשון שהצומת הנבחר הוא והשכנים שלו זה 255 או 0.ומקרה השני כאשר צומת נבחר הוא 255 והשכנים שלו הם או 255 או 0.

1. הצומת נבחר הוא 0 ולכן לפי אלגוריתם מה שרשום בחלק שמסומן כ-1 בתמונה,0 הוא קטן מ-128 ולכן תמונת פלט תקבל ערך 0,כלומר אין שינוי.עכשיו נבדוק שכנים של הצומת.אם שכן של הצומת הוא 255 ולכן צריכים להוסיף לערך הזה DIFF כפול או תלוי איפה שהשכן הזה נצמא כמו שכתוב בתמונה חלק שמסומן ב-2.אבל אם נסתכל על החישוב של ה-DIFF נראה שפשוט עושים חיסור 0-0 כך ש-DIFF שווה ל-0.מזה נובע שלא מוסיפים 0 לשכן כך שהוא נשאר להיות 255.ואם השכן יהיה 0 ובאותה דרך נוסיף 0 ולכן השכן יישאר להיות 0 כלומר גם יהיה ללא שינוי.
2. הצומת נבחר הוא 255 ולכן לפי אלגוריתם מה שרשום בחלק שמסומן כ-1 בתמונה,255 הוא גדול מ-128 ולכן תמונת פלט תקבל ערך 255,כלומר אין שינוי.עכשיו נבדוק שכנים של הצומת.אם שכן של הצומת הוא 255 ולכן צריכים להוסיף לערך הזה DIFF כפול או תלוי איפה שהשכן הזה נצמא כמו שכתוב בתמונה חלק שמסומן ב-2.אבל אם נסתכל על החישוב של ה-DIFF נראה שפשוט עושים חיסור 255-255 כך ש-DIFF שווה ל-0.מזה נובע שמוסיפים 0 לשכן כך שהוא נשאר להיות 255.ואם השכן יהיה 0 ובאותה דרך נוסיף 0 ולכן השכן יישאר להיות 0 כלומר גם יהיה ללא שינוי.

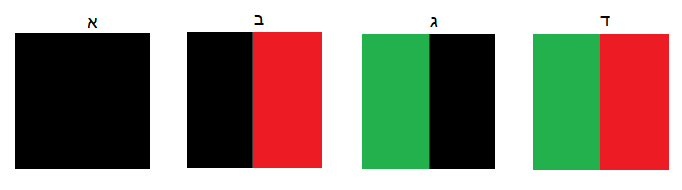
ולכן ראינו שבכל מצב שמפעילים אלגוריתם ERROR DIFFUSION על תמונה בינארית תמונה נשארת ללא שינוי.כלומר אחרי שהפעלנו פעם ראשונה אלגוריתם ERROR DIFFUSION על התמונה כך שקיבלנו תמונה בינארית אם נפעיל פעמים נוספות אלגוריתם על התמונה הזאת ,התמונה לא תשתנה כלומר תישאר בינארית כמו שהייתה אחרי פעם ראשונה שהפעלנו את האלגוריתם.

**שאלה 4:** התשובה תתבסס על דוגמה שיש בספר,פרק 6.7.3.חישוב גרדיאנט של כל רכיב בנפרד לתמונה מסוימת והשימוש בתוצאות המתקבלות ליצירת תמונה צבעונית מובילה לתוצאות סופיות שגויות.הדוגמה הפשוטה הבאה תעזור לנו להבין מדוע זה קורה.



נסתכל על ריבועים (ד) ו- (ה) שתי תמונות צבעוניות בגודל (M מספר אי-זוגי),שכל אחת מהן מורכב משלושת רכיבים (א,ב,ג) ו-(ה,ו,ז) בהתאמה.ואם נחשב גרדיאנט לכל רכיב בנפרד ונחבר אותם ביחד וניצור ריבועים (ד) ו- (ה) ואם נסתכל בנקודה אז ערך של הגרדיאנט בשני מקרים יהיה אותו דבר.אבל אם נחשוב אינטואיטיבית הגרדיאנט בריבוע (ד) אמור להיות יותר גדול יותר מאשר בריבוע (ה) מכיוון שהגבול על התמונה שכל הרכיבים עובר באותו כיוון לאומת תמונה השנייה (ה) ששם רק שניים משלושה גבולות כאלה עוברים לכיוון אחד.

ולכן אם יהיה לנו ציור הבא:



אם נחשב גרדיאנט של (ד) מכיוון שרכיב אחד הוא 0 (א) ו- (ב),(ג) 255 אבל בשני כיוונים הפוכים אז נקבל מהסכום של שלושתם שגרדיאנט שהוא שווה ל-0.אך בתמונה (ד) רואים בבירור את הגבול בין שני אזורי הצבע שונים.ודוגמה הזאת עוד פעם מראה לנו שגרדיאנט של תמונה צבעונית אינו שווה לתוצאה של יצירת גרדיאנט מסכום של רכיבים נפרדים.