

**עבודת גמר בטכנולוגיה מוכללת 5 יח"ל**

**שם בית הספר:** אורט בוסתאן אלמרג'

**שנה:** תשפ"ב – 2021-2022

**מגמה מדעית-הנדסית - התמחות אלקטרוניקה**

**נושא הפרויקט:**

שם המנחה: שדא עומרי	תואר:
שם המנחה :	תואר :
שם המנחה:	תואר:
שמות התלמידים:	
פייסל עומרי	
עבדאללה זועבי	

## תוכן עניינים

פרק 1 - מבוא	3
1.1 תיאור המצב הקיים והבעיות בו	3
1.2 הצרכים והדרישות שעבורם נידרש הפתרון	10
1.3 סקר שוק של קהל היעד	11
פרק 2 - סקר ספרות	17
2.1 עקרונות מדעיים שקיימים בפרויקט – היבטים מדעיים	17
2.2 עקרונות ופתוחים טכנולוגיים – היבטים טכנולוגיים	17
פרק 3 - סקירת מוצרים קיימים בשוק	18
פרק 4 - מחקר ופיתוח	19
4.1 תרשימים שונים לפתרון הבעיה	19
4.1.1 פתרון ראשון	19
4.1.2 פתרון שני	20
4.2 בחירת פתרון מוביל ושיקולים בבחירת הפתרון	21
פרק 5 - מימוש הפתרון	22
5.1 פרק חומרה	22
5.1.1 תרשים מלבנים	22
5.1.2 הסבר תרשים מלבנים	22
5.1.3 סימולציות שהתבצעו במהלך תכנון האב טיפוס	23
5.1.4 צילומים של הפתרון (דגם/ אב טיפוס)	23
5.2 פרק תוכנה	23
5.2.1 הסבר על שפת התוכנה שבה השתמשתם בפרויקט	25
5.2.2 תרשים זרימה של האב הטיפוס	25
5.2.3 תוכניות – הסבר על קטעי תוכנה שכתבתם	26
5.2.4 תוכנה סופית +הסברים כללים	53
ביבליוגרפיה	54

## פרק 1 - מבוא

1.1 תיאור המצב הקיים והבעיות בו  
הפרויקט שלנו עובד בתחום של המקאמאת הערבית Arabic Maqams.

הגדרת ביטויים שנשתמש בהם בפרויקט ובחוברת:

(1) מקאם – Maqam / מקאמאת – Maqams:

מקאם הוא מבנה המגדיר מסגרת ליצירה מוזיקלית במוזיקה ערבית. כל מקאם כולל סדרת צלילים אבסולוטיים ומהלכים מלודיים אופייניים לאותו סולם מוזיקלי.

שיטת המקאמאת משמשת כבסיס למוזיקה הערבית והטורקית ומבוססת ביסודה על המוזיקה הפרסית הקלאסית. יש שימוש במקאם גם בארצות הבלקן, בארצות טורקמניות במרכז אסיה, ובקווקז.

המקאם כולל סדרת צלילים אשר גובהם מוגדר על-פי המרווחים ביניהם והמהווים מסגרת ליצירה ערבית מוזיקלית. כמו במוזיקה המערבית, גם במקאם יש חשיבות לצליל הבסיס של הסולם. בנוסף, כל מקאם כולל דגשים, שהיות באזורים מסוימים בסולם הצלילים ומהלכים מסוימים המייחדים אותו, וכתוצאה מכך, ייתכנו מקאמאת שונים שהם בעלי אותם צלילים (כמו ביאת ומוחייאר או מאהור וראסט) הנבדלים האחד מהשני רק בשימוש בצלילים ובמהלכים הפנימיים.

בסולמות של המוזיקה המזרחית קיימים מרווחים הקטנים מחצי טון, ולכן שימוש ברבעי ובשמיניות הטון הוא מעשה שבשגרה, מה שמצריך מהנגן שמיעה חדה במיוחד וכלים המאפשרים חלוקת הטון למרווחים קטנים כאלה.

המוזיקה הערבית/מזרחית שמה דגש על הקשר בין המקאם בו משתמשים לבין החוויה הרגשית שהוא גורם לשומע, כאשר לכל מקאם ישנה ייחודיות שלו. בין הרגשות שנהוג לייחס למקאמאת: עצב, שמחה, געגוע, גאווה, יגון, אהבה וכו'.

(2) התפלגות המקאמאת:

(3) מספר המקאמאת (בערבית מקאמת) נרחב מאוד. הסיבה לכך היא פרמטרים שונים לקביעת מקאם, בשונה מהגדרת סולם במוזיקה המערבית שנקבע לפי היחס בין צליליו, כאן מתייחסים גם לאופן הטיפול בו (לדוגמה אם עיקר המלודיה בנויה על חלקו העליון של הסולם עשוי להיקרא המקאם בשם שונה מאילו הנגינה מתמקדת בחלקו התחתון של אותו סולם. מטבע הדברים הגדרות אלו אינן מתמטיות ובנויות על תחושה כללית). סיבה נוספת היא מספר התווים האפשרי, כיון שהמוזיקה מתבססת על חלוקה מיקרוטונאלית (רבעי ושמיניות טונים ולא רק שלמים וחצאים) מספר הסולמות האפשרי גדול יותר. מעבר לכך קיימים מקאמאת רבים שהם הרכבות של שניים או שלושה טטרקורדים מסולמות שונים. אי לכך ישנם מקאמאת המוגדרים "אבות" ולהם מספר "תולדות" ששייכים לאותה "משפחה". עם זאת מקאם מורכב שנושא אופי ייחודי, או שיש בו שימוש רב המקנה לו נוכחות בולטת יחשב גם כאב.

ישנם 7 מקאמאת ראשיים במשפחת המקאמאת, שהם:

(1) ראסט

(2) עג'ם

(3) סיקא

(4) ביאת

(5) חג'אז

(6) סבא

(7) נהוונד

(8) \*\*כורד

\*\* אבל יש גם כן את מקאם כורד שנחשב כמקאם ראשי בגלל שהוא נפוץ מאוד, לכן סה"כ נתעניין בפרויקט ב-8 המקאמא האלה.

תיאור מוזיקלי והמאפיינים של כל מקאם:

(1) ראסט: סולם המתחיל בדו, הצלילים השלישי והשביעי מונמכים ברבע טון, מקאם מרכזי במוזיקה המזרחית, נחשב למקאם שמח, משומם בקריאת קוראן באירועים חשובים כמו החגים אצל מוסלמים או בתפילת יום גומעאה – יום שישי.



דוגמא:

(2) עג'ם: סולם המתחיל על דו, משמעות המלה בערבית היא "שאינו ערבי" העג'ם נקרא גם "הסולם החזק" ואצל הערבים הוא משמש לשמחות, חתונות וחגים, גם כן לקריאת קוראן שמחה.



דוגמא:

(3) סיקא: סולם המתחיל במי שנמוך רבע טון וכולל סי מונמך ברבע טון, מקור השם בפרסית, ומשמעותו: המקום השלישי (סה-גאה). ממנו יוצא מקאם חוזאם (עם חג'אז על הסול) בו משתמשים לקריאת התורה אצל הספרדים. נפוץ מאוד בקריאת הקוראן אצל המוסלמים.



דוגמא:

(4) ביאת: מינור על רה עם צליל שני מונמך ברבע טון, מהמקאמא החשובים במוזיקה הערבית. מקור השם מהמילה בית, כיוון שנחשב למקאם הבית. מבוסס על הצליל רה עם הצליל השני מונמך ברבע טון.



דוגמא:

(5) חג'אז: רה מינור עם שנייה בימולית (נמוך חצי טון) ושלישית דיאזית (גבוה חצי טון), על שם מחוז "חג'אז" שבערב הסעודית. שימש החג'אז הוא לשירי רגש גדול, ולקריאת קוראן להציג פסוק עם רגש או כאב גדול.

דוגמא:

(6) סבא: על רה עם מי מונמך ברבע טון וסול וגם סי במולים. על פי רוב גם הרה (רק העליון) במול מה שמקנה לו חריגות גדולה, המקאם העצוב ביותר של המוזיקה הערבית. פירוש שמו הוא נעורים או תשחורת.



דוגמא:

(7) נהוונד: סולם דו מינור טבעי, נקרא על שם העיר הפרסית נהוונד (Nahawand) שבצפון פרס. אם התו סי הוא בקר הוא נקרא "נהוונד חסאס", או נהוונד חג'אז. אם התו לה הופך להיות בקר הוא נקרא "נהוונד אל כביר". יש הטוענים כי זהו מקאם של געגוע. מקאם זה היה נפוץ יותר במסורות הטורקיות מאשר בערביות, ורק בתקופה מאוחרת יותר חדר למסורת הערבית.



דוגמא:

(8) כורד: סולם שמתחיל על רה, יש את הטוענים שמקאם זה נקרא על שם הכורדים, הוא מקאם נפוץ מאוד בקריאת קוראן מהירה כך שהקורא לא מתעייף בקריאת קוראן במקאם זה.



דוגמא:

(4) בימולית: כשאומרים שתו מוזיקלי הוא בימול זה אומר שהוא מונמך בחצי טון.

- (5) דיאזית: כשאומרים שתו מוזיקלי הוא בימול זה אומר שהוא מוגבה בחצי טון.
- (6) טון: במוזיקה, טון הוא מרווח קבוע בין הצלילים, למשל המרווח בין הצליל דו, לצליל רה. טון הוא גם מונח נרדף ל"צליל".
- מרווח זה הוא כפול מהמרווח הקרוי "חצי טון", שהוא מרווח קבוע בין כל אחד מ-12 צלילי האוקטבה, כמו המרווח בין הצליל דו, לצליל דו דיאז, או המרווח בין דו דיאז לרה. במילים אחרות, כל צליל גבוה/נמוך מהצליל הסמוך לו בחצי טון.
- מבחינה פיזיקלית, כל צליל גבוה/נמוך בתדירותו מהצליל הסמוך אליו פי שורש 12 של שתיים. כלומר:
- תוספת של חצי טון (כמו מ-סי ל-דו) = הכפלה בשורש 12 של שתיים.
  - תוספת של טון (כמו מ-דו ל-רה) = הכפלה בשורש 6 של שתיים.
- (7) מרווחים: בתאוריית המוזיקה המונח מרווח מציין את היחס בגובה בין שני צלילים. במוזיקה הטונאלית, המרווח מוגדר בדרך כלל כיחס בין התדירויות של שני הצלילים [1] או על ידי המרחק בין התווים המייצגים אותם על הסולם הדיאטוני.
- היחידה הבסיסית בה נמדד המרווח המוזיקלי הוא ה"טון". היחידה הקטנה ביותר במוזיקה המערבית היא חצי טון, ולפיה מכוונים כלי הנגינה המערביים. במוזיקה מזרחית לסוגיה משתמשים גם ברבעי טונים ובסולמות שונים.
- (8) סולו - solo: הוא מונח המתאר פעילות הנעשית על ידי בו אדם אחד בלבד, נתעניין בסולו של קורא קוראן אחד או זמר אחד.
- (9) נשאז: כיוון לא נכון או זיוף.
- (10) סולם: סולם מוזיקלי משתרע על פני אוקטבה ומחלק אותה לתווים. אוקטבה היא מרווח בין שני צלילים שתדירותו של הגבוה יותר כפולה בדיוק מתדירותו של הנמוך. למשל אוקטבה שמתחילה בתו לה שתדירותו 440 הרץ מסתיימת בלה גבוה יותר שתדירותו 880 הרץ. המוזיקה המערבית מבוססת על חלוקת האוקטבה ל-12 חלקים שמכוונים חצאי טונים.
- (11) אירטקאז: הוא התו המוזיקלי של תחילת הסולם.
- (12) תדר: בפיזיקה, המונח תדירות (או תדר) של תופעה מחזורית מציין את מספר המחזורים שמתבצעים בכל יחידת זמן. דוגמה לכך היא גוף קשיח שמסתובב בחופשיות - תדירותו היא מספר הסיבובים שהוא מבצע בכל פרק זמן קבוע. את התדירות נהוג לסמן ב-f והיא נמדדת במערכת היחידות הבינלאומית בהרץ (Hz), כאשר הרץ אחד הוא מחזור אחד לשנייה. לדוגמה, זרם חילופין: תדר הרשת החשמלית בישראל הוא 50 הרץ, כלומר המתח החשמלי משתנה במחזורים בקצב של 50 מחזורים בכל שנייה.
- (13) Sample rate: תדירות הדגימה (Sampling rate, נמדד לרוב ב-KHz) מציין את מספר הפעמים בשנייה שהצליל נדגם. ככל שהתדירות תהיה גבוהה יותר כך הצליל יהיה קרוב יותר למקור.
- (14) Cents: סנט היא היחידה המקובלת ביותר להשוואה בין מרווחים מוזיקליים וייצוגם. הסנט היא יחידת מדידה הנובעת מחלוקה לוגריתמית של חצי הטון המשווה ל-100 חלקים שווים. 1200 סנטים שווים לאוקטבה אחת ו-100 סנט שווים לחצי טון בכוונן מושווה.
- תיאור מצב קיים:
- במצב הקיים נרצה לגלות את המקאם המשמש בקטע קול של קראת קוראן.
- למה קריאת קוראן: קריאת קוראן היא קראיה שבה יש רק קול אחד והוא של הקורא, ובכך אנו מקבלים את הקול כסולו – solo, ניתן לראות בהמשך החוברת שהאלגוריתם יעבד לא רק על קבצי קריאת קוראן אלא גם כן על קטעי סולו של כלי נגינה או זמרים, אבל בניסיונות שלנו לאלגוריתם ולקוד אנו מתעניינים אך ורק בקריאת קוראן.

למה קריאת קוראן: כי קריאת קוראן הוא מה שמוסלמים עושים לפחות 5 פעמים ביום, וזה במהלך התפילות, וההשתמשות במקאמאת במהלך קריאת הקוראן נחשבת מהסיבות הכי חושות כדי לציין איזשהו קורא ברמה ספציפית, לכן, קורא שקורא קוראן עם שלמות בהשתמשות במקאמאת ייחשב כקורא מצטיין וחזק, כי אנשים שמתפללים מאחרו יהיו לו יראת כבוד וזה משהו שחשוב מאוד לנו כמוסלמים, בנוסף לכך שהנביא מוחמד הורה לנו לשיר בקוראן, ומסבירים הסבירו את ההוראה הזאת שלשיר זה שהקורא ינסה לשפר כמה שאפשר את הקול שלו בקריאת קוראן, ולהשתמש במקאמאת זה נחשב כשיפור מאוד גדול בקול ובכך תהי לקורא קריאה יפה.

הבעיות במצב הקיים:

יש כל מיני בעיות בהשתמשות במקאמאת, הם:

- (1) נשאל: אחת הבעיות העיקריות בהשתמשות במקאמאת היא שהשתמשות בלי לשכלל את הקריאה תגרום לכך שיהיה כמות נשאל מאוד גדולה, וזה יגרום לכך שהמתפללים מאחרו של הקורא יאבדו את יראת הכבוד שלהם, וכשהמקאם שמשמששים בו יהיה יותר קשה, זה יגרום למי שלא ישכלל את הקריאה שיהיה לו כמות יותר גדולה מנשאל וזאתי הבעיה העיקרית לאלה שמשמששים במקאמאת בלי ללמוד אותם.
- (2) מיומנות: מיומנות נחשב כחלק חשוב שיצטיין בו קוראים שנחשבים ברמה מעל קוראים אחרים, ומי שאין לא מיומנות בהשתמשות במקאמאת, לא יגיע רחוק כקורא מצטיין, בנוסף לכך שרוב המקאמאת שהמקור שלהם לא ערבי, יותר קשה לקוראים ערבים מוסלמים להיות מיומנים המקאמאת האלה בגלל שהם רגילים יותר לשמוע ולקרוא מקאמאת שמקורם ערבי, ומכיוון שמספר המקאמאת הערביים יחסית קטן ממספר שאר המקאמאת, אזי כדי לקרוא קוראן באופן יפה ושיהיה אינטונציה לקוראן בשיטה אהובה ומושכת, יהיה למיומנות חלק גדול במיון הקוראים.
- יש כל מיני סוג של בעיות שלא נתיחס להם הפתרון שלנו אבל חשוב לציין אותם:
- (3) השתמשות במקאם במקום הלא נכון בקריאת קוראן: ידוע כי כל פסוק או סיפור או דיאלוג בקוראן, יתאים לו אופן קריאה כלשהו שמתאים לתוכן שלו, לכן, ניתן לראות לפי כך שקורא שיש לו מיומנות גבוהה בקריאת קוראן ובהשתמשות במקאם ואין לא נשאל בכלל, ייתכן שהמתפללים מאחרו לא ירגישו את כל זה בגלל שהוא פשוט לא משתמש במקאם הנכון, השתמשות במקאם הלא נכון לא אומרת שלכל פסוק צריך את המקאם x או y אך ורק, אלא זה אומר שלמשל לא יתאים לקרוא פסוק שמדבר על נביא שעם שלו היו עונים בו במקאם כמו עגאם שנחשב כהמקאם הכי שמח שיש, עוד דוגמה, שאי אפשר לדבר על איך שהמאמינים בנביא יכנסו לגן העדן, ולקרוא את הפסוק במקאם סבא שנחשב כהמקאם הכי עצוב שיש, בסוף, בחירת המקאם תהיה מתבוססת על הקורא עצמו ועל הרגשתו.
- (4) התייחסות למקאם כמשהו שיותר חשוב מהתגיד - Tajwid של הקוראן: בזמן הזה, יש סוג של אנשים שקוראים את הקוראן בהתייחסות אך ורק למקאם שהם קוראים בו והמיומנות שלנם בקריאה, והם מדלגים על חוקי הקריאה של הקוראן שנקראים תגיד בערבית, חלק זה מאוד חשוב כי יש סדר עדיפות בדת, וקריאת קוראן זה לא משהו פשוט בדת, ואי אפשר לסלוח לקוראים כאלה, כי הפיכת סדר עדיפות בלימודי הקוראן תגרום לבלגן בין אנשים, וכמו שהמעדנים המוסלמים הסכימו על כך שלמידת תגיד מאוד יותר חשובה מהמקאמאת.

תיאור הרעיון הכללי לפתרון הבעיה:

נתייחס לשתי הבעיות הראשונות שזכרנו בפסקה קודמת שהן בעיית הנשאל ובעיית המיומנות:

הרעיון שלנו לפתור את שתי הבעיות האלה יהיה באופן הבא:

נבנה אלגוריתם וקוד שיקבל בתוך קובץ הקלטה של איזשהו קטע קריאת קוראן, והאלגוריתם יחזיר למשתמש כמה נתונים:

- (1) את המקאם שהוא משתמש בו.
  - (2) את כל המקאמאט שהוא עבר בהן בקריאה שלו, ואחוז כל מקאם מבין כל זמן הקריאה שלו.
  - (3) תיאור את התווים המוזיקליים שהוא עבר בהן, ומספר ההופעות שהוא עבר בהן.
  - (4) לתאר לו את התווים שהמקאם שהשתמש בו צריך לעבור בהן, ולהראות למשתמש על אילו תווים הוא עבר ויש בהן הופעה אבל לא צריך שיעבור בהן.
  - (5) ולפי סעיף קודם אנו יכולים לחשב את אחוז הנשאז של הקורא, ונחזיר לו את אחוז הנשאז שיש לו.
  - (6) לתת למשתמש כמה אופציות לבחירה כדי לשפר את האלגוריתם, כמו האירתקאז שלו.
- הרעיון הכללי לפתרון הוא כבא:

- (1) להעביר את הקובץ של הקריאה שנקבל למערך של תדרים לפי sample rate מסוים שיהיה סטנדרטי, כלומר אם מקבלים קובץ של 15 שניות עם sample rate 44khz אזי יש לנו  $15 \times 44 = 660000$  תדר בקובץ, נעשה המרה לפי sample rate חדש שהוא למשל 800hz ואז נקבל 825 הדגמה לתדרים.
- (2) בשלב הבא נצטרך להתאים את כל תדר מבין התדרים שקיבלנו במערך מהשלב הקודם לתוו מוזיקלי שמתאים לו לפי הטבלה הבאה:

	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

אבל בגלל שיש מקאמאט שיש בהן רבע טון, אז לכל תוו מוזיקלי נחשב את ההגבה ברבע טון שלו לפי המשוואה הבאה, כדי להגביה רבע טון אזי נעשה תדר + 50cents לכן המשוואה תהי:

$$Note_{frequency} \times 1.00057778950655^{50}$$

ובכך נקבל את התדר של התוו שגבוה ברבע טון.

- (3) אחרי שהתאמנו כל תדר שהופיע בקובץ, נחפש את מבנה ההופעות של התווים המוזיקליים שנמצאים במערך, ולפי כך נחליט איזה מקאם זה, נעשה זה על ידי לולאה לאורך המערך של התווים המוזיקליים, נתחיל לחפש את האירתקאז אם הוא לא היה נתון, וזה על ידי חיפוש של התוו הראשון שמופיע במערך, ולאחר מכן נחפש עבור המרווחים שמתוארים בטבלה הבאה, אם איזשהו מקאם מתאים למבנה של ההופעות, אזי נחזיר אותו כתשובה, אחרת נחזיר שגיאה, אם האירתקאז היה נתון אזי נחפש אך ורק ביחס לאירתקאז שהמשתמש נתן. טבלת המרווחים בין תווי כל מקאם:

תוו	תוו	תוו	תוו	תוו	תוו	תוו	תוו	תוו
שמיני	שבעי	שישי	חמישי	רביעי	שלישי	שני	ראשון	המקאם
0.75	0.75	1	1	0.75	0.75	1	0	ראסט
1	1	0.5	1	1	0.75	0.75	0	ביאת
1	1	0.5	1	0.5	1.5	0.5	0	חיגאז
1	0.5	1	1	1	0.5	1	0	נהונד
1	1	0.5	1	1	1	0.5	0	כורד
1	1	0.5	1.5	0.5	0.75	0.75	0	סבא
0.75	1	0.5	1.5	0.5	1	0.75	0	סיקא
0.5	1	1	1	0.5	1	1	0	אגאם

4) לפי המקאם והאירטקאז שלו שמצאנו בשלב קודם, נחפש את מספר ההופעות של כל מקאם לפי התווים שלו ביחס לאירטקאז, ואז נחשב את אחוז הופעת כל מקאם על ידי חישוב אחוז מספר ההופעות של התווים המוזיקליים שלו ביחס למספר ההופעות הכללי של התווים המוזיקליים.

חלוקת העבודה:

נושא	תיאור צמיתי	חלוקת עבודה	רלוונטיות לפרויקט
מוצרים דומים	- האם יש אפליקציות או מחקרים דומים? - מה החסרונות במוצרים הדומים?	עבדאללה	אנחנו חייבים לדעת אם יש מוצרים דומים שכבר עונים על בעיה דומה
רעיון אלגוריתם	- איך לעשות כל סוגי ההתאמות בין קובץ הקול למקאמאת? - איך לבדוק את סוג המקאם?	פיסל	אנחנו נשתמש באלגוריתם הזה כדי לפתור את הבעיה המרכזית שלנו כפי שהוסבר קודם
ארדינו וקבלת קול ממיקרופון	- מה זה ארדינו ואיך עובד - מה זה מיקרופון - איך אפשר לשלב בין ארדינו לבין מיקרופון - מה התוצאה שנקבל מהקוד של הארדינו	עבדאללה	החלק של האלקטרוניקה כדי לחבר את הקול מהאלקטרוניקה למחשב דרך הארדינו
כתיבת האלגוריתם של גילוי המקאם ב-MATLAB	- כתיבת הקוד לפי האלגוריתם - לסדר את הקוד להוסיף רעיונות כדי להקל את הסיבוכיות להוסיף interface קל למשתמש	פיסל	מימוש האלגוריתם ב-MATLAB שזה בעצם הפתרון שלנו, זה החלק של כתיבת הקוד שיפתור את הבעיה
שילוב MATLAB וארדינו	- איך להעביר את הקול שמקבלים ממיקרופון לתוך קובץ כנדרש ב-MATLAB - איך אפשר לוודא תקינות ואי סטייה בין המיקרופון והארדינו לבן הקובץ שמגיע ל-MATLAB	עבדאללה	החלק הסופי של כל הפתרון שבו משלבים בין הקוד של ה-MATLAB לבין הארדינו ובכך מסיימים את המימוש של הפתרון באופן שלם
בדיקת תוכנה,	- הרצת הקוד על כל מיני קבצי קול של	פיסל	כדי לוודא שהפתרון עובד בצורה נכונה כנדרש, ושיהיה



לנו מוצר סופי שעונה על הבעיה ופותר אותה באופן הכי מסודר.		קראית קוראן לפי המקאם ובדיקת התשובות המוחזרות מהקוד - לוודא שהקוד שנכתב מסודר ועובד על כל סוגי הקול כנדרש	וודאות אלגוריתם נכון
--	--	--	----------------------

## 1.2 הצרכים והדרישות שעבורם נדרש הפתרון:

הפתרון הזה מהווה ומספק הרבה צרכים שהרבה אנשים היו מחפשים במהלך השנים הקודמות, ולעומת ההתקדמות הטכנולוגית וההשתפרות של AI אבל הצרכים האלו לא הסתפקו עד עכשיו, הצרכים והדרישות שיספקו על ידי הפרויקט שלנו לפי המחקר שיוצג בשלבים מתקדמים מחוברת זאת הם:

- (1) יהווה לאנשים ולמתעניינים בתחום המקאמאט, לשפר את הממומנות שלהם בתחום זה על ידי שיפור הקול והביצוע שלהם בקריאת קוראן, וזה יתקבל על ידי כמה שלבים:
  - (i) שלב ראשון זה להגיע למקום שבו המבצע של הקריאה הקוראנית ינסה להשתייך ולהישאר דבוק למקאם אחד למהלך כל קריאתו, ואז בכל בדיקה באלגוריתם הוא יכול לדעת האם הוא באמת הולך לכיוון הנכון והאם הקריאה שלו היא נשמרת על אותו מקאם, ואם כן אז עד שיגיע לשלב שבו יותר מ-98% מקריאתו היא על המקאם שהוא בחר ברצונו, ואז הוא יעבור לשלב הבא.
  - (ii) לבדוק כמות הנשאז שלו על ידי הנתונים שהאלגוריתם מחזיר, ולוודא שאחוז המקאם שמשתמש בו הוא כמעט כפול מאחוז כל מקאם אחר שהנתונים מציגים, ולפי כך הוא יוודא שכמעט אין לו שום נשאז בקראות שלו.

ניתן לראות שהפתרון בדרך כלל מיועד לאנשים שקוראים קוראן ומתעניינים להתמתחות במקאמאט, או אנשים שרוצים רק לדעת באיזה מקאם הם קוראים הקוראן, אבל בגלל שהאלגוריתם כפי שהוסבר קודם, מהווה הרבה נתונים ואנליזציה על קטעי הקריאה האלה, לכן מי שיהיה לו הכי טוב מהפתרון הם השייחים המוסלמים כי בדרך כלל הם האנשים שאופן קריאתם לקוראן מאוד משפיע, כי כל הזמן אנשים עוקבים אחריהם בתפילה וזה יגרום להם להתעניין בכך שישאירו ואפילו להעלות את מספר המתפללים במסגדים על ידי משיכת האנשים להתפלל אחרי אימאם (הראשון בתפילה שאחרים עוקבים לו) ולהתקרב לאלוהים.

1.3 סקר שוק של קהל היעד  
עשינו מחקר כדי לבדוק שבאמת קהל היעד מתעניינים בפתרון לבעיית גילוי המקאמאת שהם משתמשים בהם, במחקר שאלנו ארבע שאלות כלליות:

(1 באיזה רמה אתה מתאר את הידע שלך במקאמאת? אופציות:

- i. מומחה
- ii. מקצועי
- iii. בינוני
- iv. פחות מבינוני
- v. לא יודע בכלל

(2 האם אתה משתמש או מנסה להשתמש במקאמאת כשאר אתה קורא את הקוראן? אופציות:

- i. תמיד
- ii. רוב הזמן
- iii. לפעמים
- iv. בכלל

(3 האם אתה מעוניין לדעת את דרגת המיומנות שלך בהשתמשות במקאמאת? אופציות:

- i. כן, באופן גדול
- ii. כן, באופן בינוני
- iii. לא

(4 באם אתה מעוניין לדעת מה המקאם שאתה קורא? אופציות:

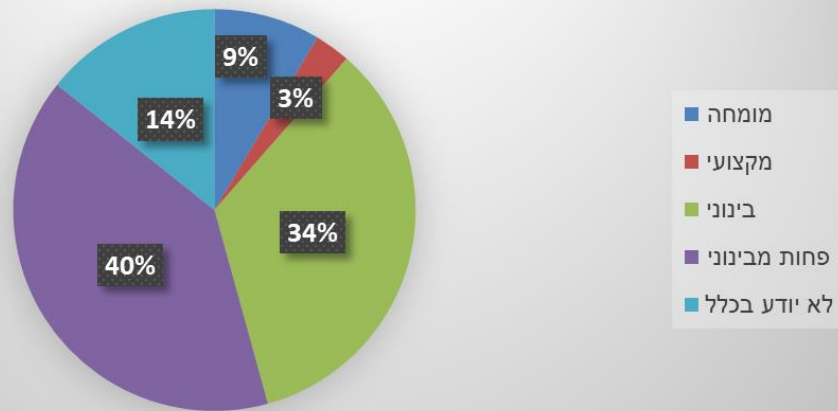
- i. כן, באופן גדול
- ii. כן, באופן בינוני
- iii. כן, אבל לא כל כך
- iv. לא

גם כן היה אחרי השאלות מבחן קצר מאוד שבודק את הידע של האנשים שענו במקאמאת.

בכל שאלה היה מצורף קטע של סורת אל-פאתיחאה, כל פעם במקאם אחר, המקאמאת היו מקאמאת שרוב הזמן משתמשים בהם לקריאת קוראן או לקריאה לאזאן בכל המסגדים באזורי הארץ. שהם מקאם כורד, חיגאז וראסט.  
התשובה של שאלה 5 היא כורד.  
התשובה של שאלה 6 היא חגאז.  
התשובה של שאלה 7 היא ראסט.

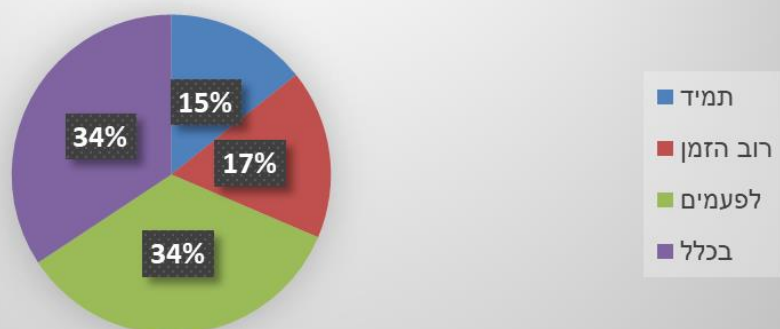
ולאחר שענה 35 אנשים על המחקר הזה, כך שרוב האנשים הם אימאם של איזשהו מסגד, או אנשים התמחים בדת, התוצאות שקיבלנו היו מדהימות:  
שאלה 1:

## באיזה רמה אתה מתאר את הידע שלך במקאמאת?

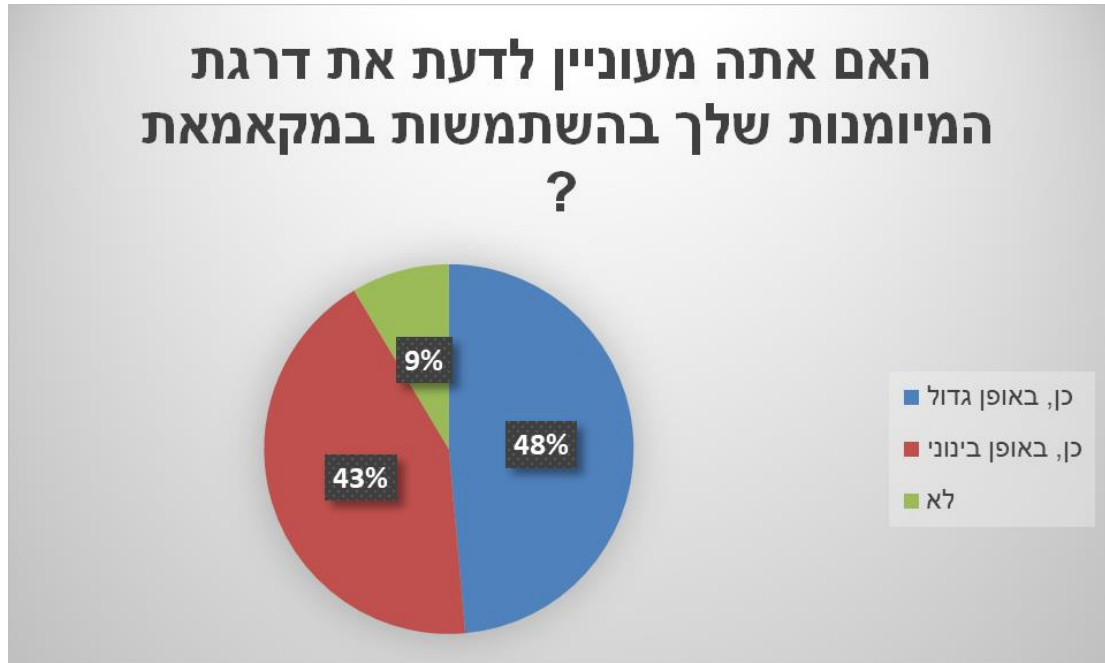


שאלה 2:

## האם אתה משתמש או מנסה להשתמש במקאמאת כשאר אתה קורא את הקוראן?

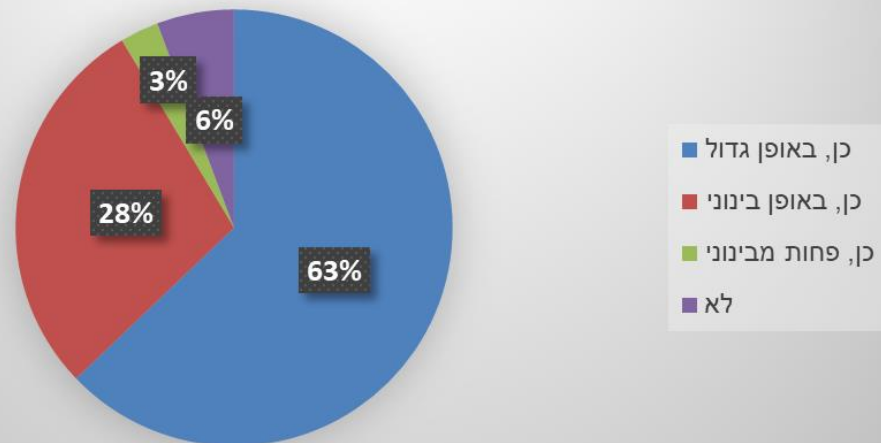


שאלה 3:



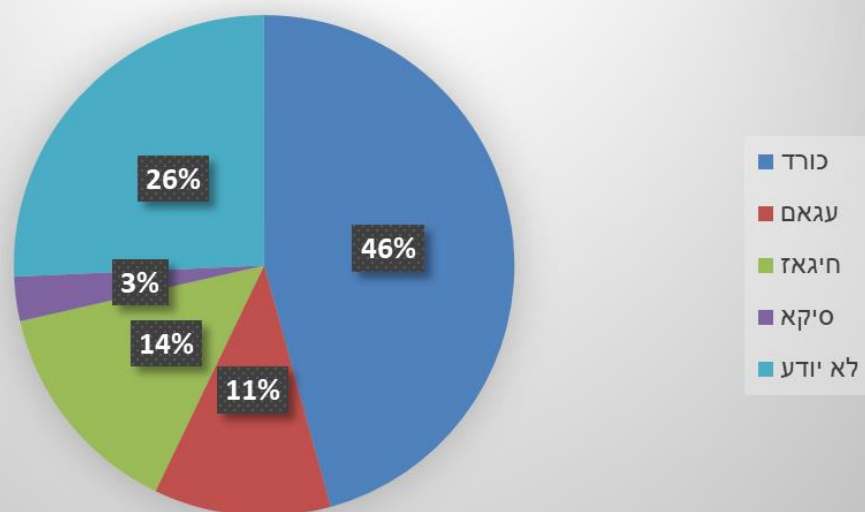
שאלה 4:

## האם אתה מעוניין לדעת מה המקום שאתה קורא?



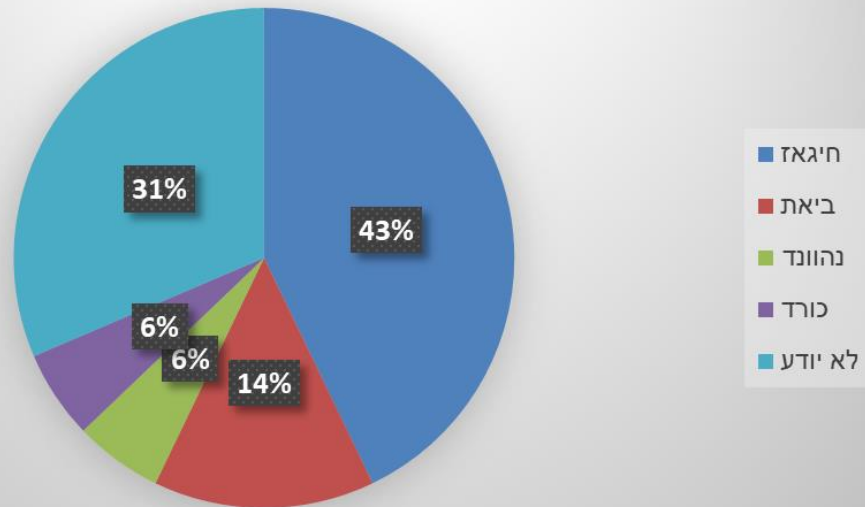
התשובות לשאלה 5:

## שאלה 5



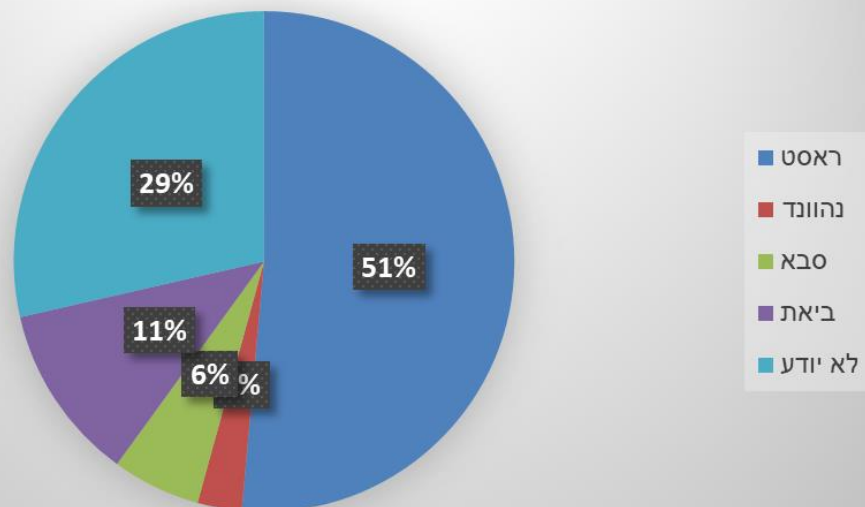
התשובות לשאלה 6:

## שאלה 6



התשובות לשאלה 7:

## שאלה 7



ניתוח סקר:

ניתן לראות מהתשובות על הסקר שרוב האנשים שענו הודיעו על כך שהם לא מקצועיים באחוז 54% ואלה שהם מקצועיים או מומחים האחוז שלהם הוא רק 12%! לאחר מכן ניתן לראות שאלה שמשתמשים במקאמאט או אפילו מנסים להשתמש הם 64% וזה אחוז נראה גדול כי רק 34% לא מנסים ולא משתמשים במקאמאט. לאחר מכן בשאלה 3 ניתן לראות ש 91% מהאנשים מעוניינים לדעת את דרגת המיומנות שלהם במקאמאט וזה אחוז גדול מאוד. ובשאלה 4 ניתן לראות ש 94% מהאנשים רוצים לדעת את המקאמ שהם קוראים בו, וזה חוזר לכך שכמעט לכל אדם יש איזשהו מקאמ שהוא תמיד משתמש בו לא משנה האם הוא יודע או לא זה כמו משהו שנולד בו.

אחר כך לחלק השני שלל השאלות אחוז אלה שהצביעו שהם לא יודעים את המקום במהלך השאלות 5, 6 ו 7 הוא 26%, 31% ו 29% בהתאם וזה מראה כמה גדולה כמות האנשים שהם רוצים לדעת ומתעניינים במקאמאת אבל הם לא יודעים. בצד שני ניתן לראות שבשאלות 5, 6 ו 7 אחוז אלה שענו באופן נכון הוא 46%, 43% ו 51% בהתאם וזה סותר את הטענה שאף אחד מהם לא יודע את המקאמאת אלא יש אחוז טוב של אנשים שיודעים את המקאמאת אבל לא מספיק בהתייחסות שכל אחד מהם הוא אמאם של מסגד וקורא קוראן רוב הזמן.



## פרק 2 - סקר ספרות

### 2.1 עקרונות מדעיים שקיימים בפרויקט – היבטים מדעיים

- (1) חשמל ואלקטרוניקה: בפרויקט השתמשנו ברכיבים שנדרשנו להכיר כדי להשתמש בהן, כמו כרטיס ארדינו, מיקרופון, הרכבת מעגל חשמלי עם שילוב בין נגדים, הרכבת מקור אור קטן במעגל.
- (2) תקשורת נתונים: המערכת שבנינו הינה מערכת תקשורתית בכדי להעביר את המתונים מכרטיס הארדינו ומהמיקרופון לתוכנה שהיא MATLAB.
- (3) המרת תדרים לתווים מוזיקליים: השתמשנו באלגוריתם בעובדה שכל תדר מתאים לתוו מוזיקלי כלשהו.

### 2.2 עקרונות ופתוחים טכנולוגיים – היבטים טכנולוגיים

- (1) פורמט קובץ wav: נדרשנו ללמוד את סוג הפורמטים ומבני הנתונים של פורמט קובץ wav, ואיך המידע של. הקובץ נשמר לפי sample rate.
- (2) המרת תדרים לתווים מוזיקליים: השתמשנו באלגוריתם בעובדה שכל תדר מתאים לתוו מוזיקלי כלשהו.

### פרק 3 - סקירת מוצרים קיימים בשוק

לפי חיפוש וסקירת מוצרים דומים שקיימים בשוק, לא מצאנו שום אפליקציה שפותרת את הבעיה, ואפילו שום אפליקציה שמנס לגלות את המקום עבור איזשהו קובץ קול, אלא מצאנו שני מוצרים דומים:

- (1) אפליקציה לגילוי תוו מוזיקלי וביחס לתדר שקולטת מהמיקרופון:  
מצאנו כמה אפליקציות כאלה, אפליקציות כאלה משתמשים בהם בעיקר לרעיון אחד, והוא כדי לאזן ולעשות tune לכלי נגינה, כמו כינור, עוד, וגיטרה וכו".  
למשל כמו האפליקציה DaTuner שנמצאת בחנות אנדרואיד, אבל עדיין אפליקציה זאת לא יכולה לפתור ולגלות את המקום המשומש.
- (2) מחקר מדעי שנעשה באוניברסיטת שארגה באיחוד האמירויות הערביות, המחקר הוא בשם:  
Classifying Maqams of Qur'anic Recitations using Deep Learning  
במחקר הזה מיוצג רעיון ושיטה מאוד מתקדמת בתחום ה-Al שמשמשים בו במחקר, גם כן במחקר הזה משתמשים בהרבה חוקים של הקול ושל התדרים וכל מיני סוגי אנליזה שמאוד מתקדמת לגילוי קטעי קול שאפילו מורכבים מיותר ממקום אחד, החסרונות המחקר זה הם:  
i. בחוברת המחקר השתמשו רק בדוגמא אחת לקטע קול שבו יש קריאת קוראן, והאנליזה שעשו על הקטע הזה שהסבירו עליה, לא בוודאי תעבוד על קטעי קול אחרים, אפילו ניתן לראות שהאנליזה מתעניינת אף ורק בקטע אחד.  
ii. אין אלגוריתם לרעיון ואין מימוש, וזה אומר שמחקר לא הגיע לשלב שבו הוא באמת מספק את הדרישות שהסברנו קודם.

פרק 4 - מחקר ופיתוח

## 4.1 תרשימים שונים לפתרון הבעיה

בכל פתרון צריך לצרף תרשים מלבנים, הסבר מילולי איך עובדת המערכת ויתרונות וחסרונות של המערכת המוצגת בפתרון

## 4.1.1 פתרון ראשון

הפתרון הראשון שהתחלנו בו, הוא לפי הקוד, מתקיים כאשר  $\text{Filtering} = 0$ , הפתרון מיועד עבור קטעי קול בלי שום נשאז אלא רק קול שבא מכל אלקטרוני ושעובד רק בטווח שבו לא יוצאים משום תו מהתווים השייכים למקאם, מה שהפתרון עושה הוא סינון תדרים באופן שבוא לא מתעניין במספר הופעת כל תדר, הוא רק מתייחס למספר ההופעות כ "יש" או "אין" כלומר אם תדר מסוים הופיע או לא, ואז אחר כך מכניסים את התדרים שהופיעו ללולאה שעושה התאמה, כל תדר יותאם לתו המוזיקלי שהתדר מציג לפי טבלת התאמות, למשל אם יש לנו את קטע התדרים הבא:

146.83,155.56,185,196,220,233.08,261.63

ההתאמה תהיה באופן הבא:

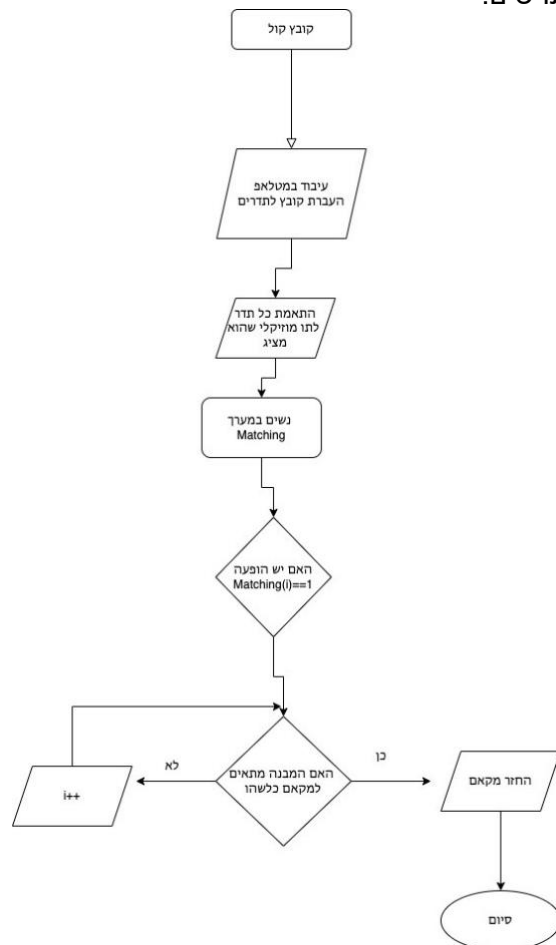
$D_2, D\#2, F\#2, G_2, A_2, A\#2, C_3 D_2, D\#2, F\#2, G_2, A_2, A\#2, C_3$

ואז, אנו יכולים לגלות את המקאם לפי הנתון הבצורת תווים מוזיקליים, וזה דרך פונקציה שתקבל את מערך התווים שהופיעו, ומחזירה את המקאם שמשמש בקטע קול הזה, למשל בדוגמא הזאת המקאם הוא ח'גאז - Hijaz.

יתרון: יעבוד באופן מהיר כן סיבוכיות נמוכה.

חסרון: לא יעבוד למטרה הרצויה.

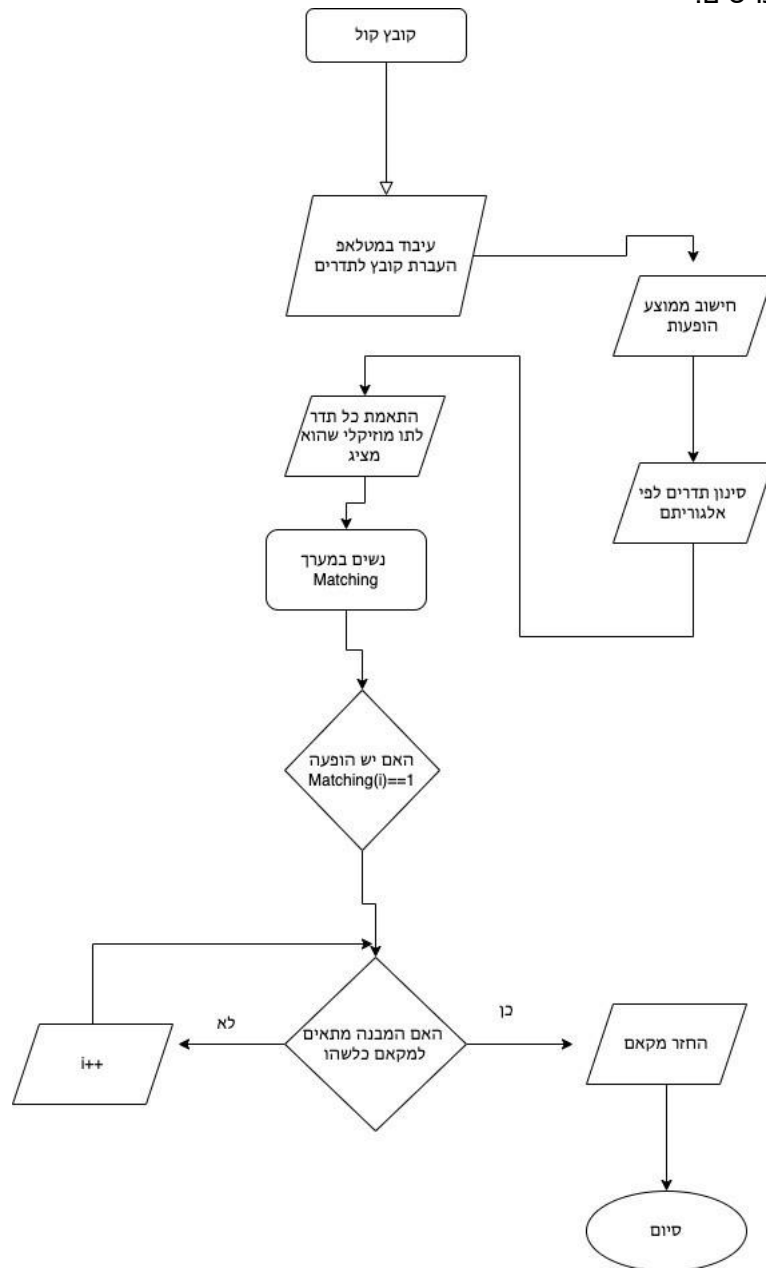
תרשים:



## 4.1.2

## פתרון שני

הפתרון השני הוא שיפור מאוד גדול לפתרון שני שבוא משפרים את השלב שבוא מבצעים סינון לתדרים, בשלב זה לא נתייחס לכל הופעה כהופעה, אלא נלמד קצת על מערך התדרים כדי לסנן אותו מתדרים לא רצויים, מה שהוא עושה הוא לחשב את ממוצע מספר ההופעות, ואחר כך נעבור על מערך ההופעות של התווים המוזיקליים ואז אם יש איזשהו תוו שמופיע פחות מתו אחר נתעלם ממנו ונאפס אותו ונסמן שהוא לא הופיע, ובכך פתרנו את הבעיה שאם יש קצת נשאז בקובץ, שהנשאז הזה יגרום לתוצאות ונתונים לא נכונים המוחזרים על ידי הקוד. ואז נמשיך לשלב שבוא נחפש את מבנה המקום המשומש. יתרון: עובד כנדרש לקבצי קול של זמן לא קצר, הכי טוב שיש. חסרון: סיבוכיות גדולה, יכר זמן כדי לרוץ, בנוסף שלא עובד על קטעי קול מאוד קצרים כי רוב ההופעות יהיו פחות מהממוצע כי לא עוברים הרבה על תווים מוזיקליים. תרשים:



#### 4.2 בחירת פתרון מוביל ושיקולים בבחירת הפתרון

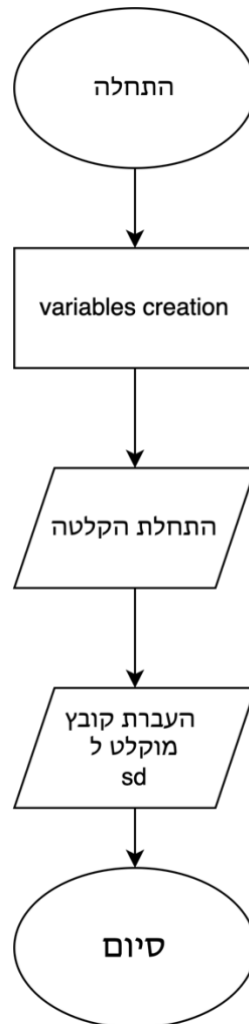
הפתרון המוביל שנחבר כדי שיהיה הפתרון הראשי, הוא הפתרון השני אבל נשאר אופציה למשתמש לבחור פתרון ראשון על מנת לספק את קטעי הקול הקצרים, הבחירה זאת התבססת על כך שכמעט כל קבצי הקול שהרצנו בפתרון השני עבד כמו שצריך, ואם היה הרבה נשאז בקטע אז הייתה לנו שגיאות כי זה מה שצריך להחזיר כי יש נשאז, אבל המקרים שבהם היה מיומנות הפתרון פשוט עבד כנדרש וסיפק את כל הדרישות.

## פרק 5 - מימוש הפתרון

### 5.1 פרק חומרה

#### 5.1.1 תרשים מלבנים

תרשים מלבנים של הפתרון הנבחר והסבר מילולי מפורט על פעולת האב טיפוס.



#### 5.1.2 הסבר תרשים מלבנים

כמו שמתואר בתרשים המלבנים, אנו פשוט כמו שנסביר בשלבים הבאים, נתחיל הקלטה עבור הפלט מהמיקרופון לתוך קובץ, כשנסיים ההקלטה הקובץ שהוקלט פשוט יועתק ל SD card המצורף בכרטיס הארדינו.

5.1.3 סימולציות שהתבצעו במהלך תכנון האב טיפוס עשינו כל מיני סוגים של סימולציות כדי לוודא תקינות של רכיבי חומרה, מה שהיה הכי חשוב לבדוק זה תקינות ודיוק של קליטת התדרים של המיקרופון, אחרי שווידאנו שהמיקרופון עובד תקין, התחלנו לנסות לשפר את הקוד של הארדינו, הייתה בעיה בקבצים עם הקלטה של יותר מדקה, אבל אחרי שגילינו שהבעיה הייתה ב sample rate שנקבע, ואחרי קביעת אחד חדש, ניסינו כמה סימולציות של קריאות קוראן עם הפעלת ההקלטה ברכיבים, ואז זה עבד תקין עבר sample rate ששווה ל16KHz.

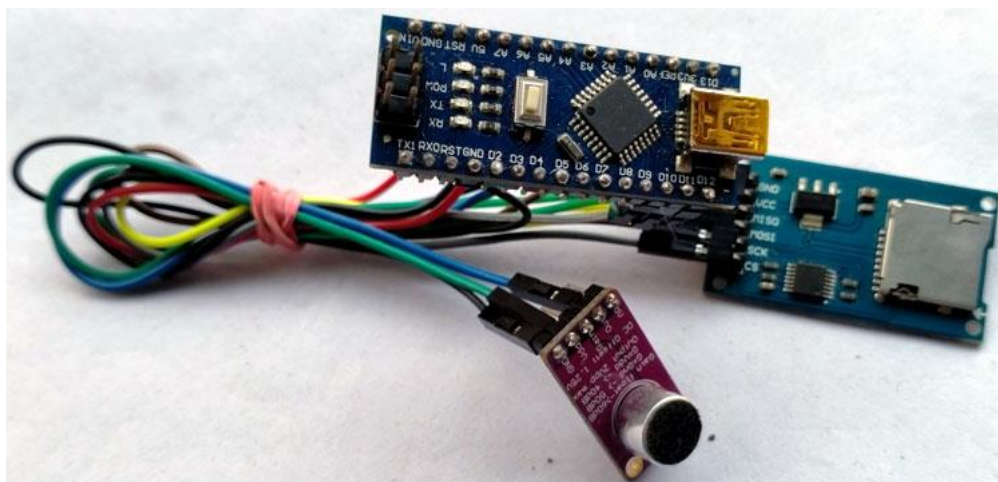
עוד בעיה הייתה בעת העברת הקובץ שהוקלט מה ארדינו לSD card, הבעיה הייתה שהסנכרון לא היה מתבצע באופן נכון בין ההקלטה לSd card, ואחרי כמה סימולציות וגילוי שגיאה בקוד, תיקנו אותו.

5.1.4 צילומים של הפתרון (דגם/ אב טיפוס) תיאור של התהליך של מימוש הפתרון, שיקולים החלטות.

המיקרופון מדגם: MAX9814.



הרכיב:

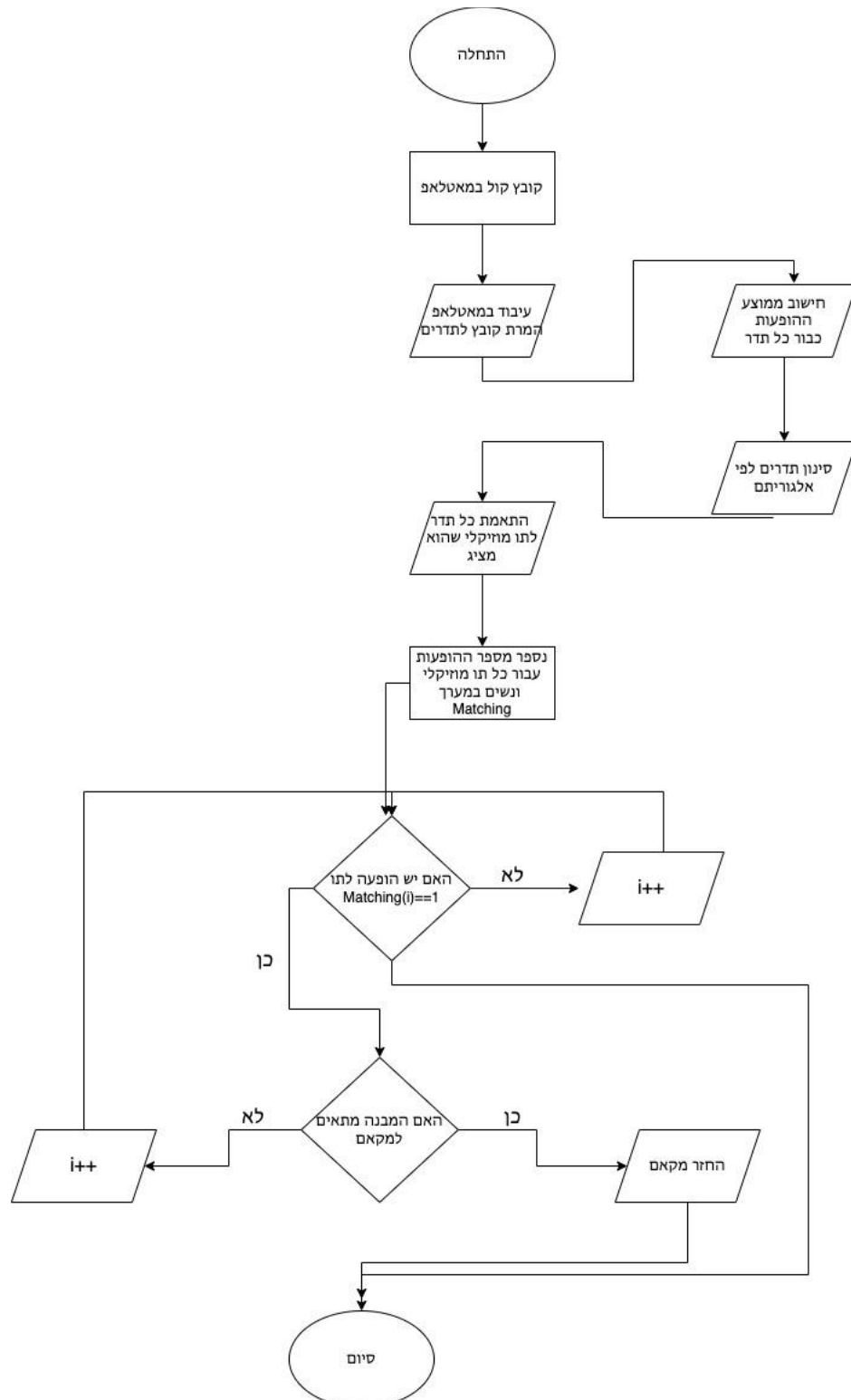




## 5.2 פרק תוכנה

5.2.1 הסבר על שפת התוכנה שבה השתמשנו בפרויקט .  
 בפרויקט זה השתמשנו בשתי שפות כדי לממש את הקוד, לפי שהוסבר קודם, שיש לנו שני חלקים בקוד, אחד של Arduino כדי לחבר המיקרופון עם התוכנה, והשני הוא קוד בשפת MATLAB שמימשנו בו את הקוד שמגלה את ה"מקאם" של קטע הקול שהוקלט בעזרת המיקרופון.

## 5.2.2 תרשים זרימה של האב הטיפוס .



## 5.2.3 תוכניות – הסבר על קטעי תוכנה שכתבתם .

Arduino:

(1)

```
#include <TMRpcm.h>
#include <SD.h>
#include <SPI.h>
```

טעינת ספריות לקוד,

ספריית TMRpcm משתמשים בה כדי לשלוט בפקודות עם קבצי wav הנשמרים בsd card כמו הפעלת איזשהו קובץ אודיו השמור בתוך הsd.

ספריית SD משתמשים בה כדי לשמור ולטעון כל מיני סוגים של קבצים לתוך הSD שמחובר לכרטיס הארדינו.

ספריית SPI תאפשר לנו לעשות communication בין מכשירים, בקוד שלנו, תאפשר לנו להתחבר עם המחשב והSD.

(2)

```
TMRpcm audio;
int file_number = 0;
char filePrefixname[50] = "record";
char exten[10] = ".wav";
```

בשורות האלה נייצר את המשתנים שנשתמש בהם, קובץ audio והprefix של השם של הקובץ שנקליט בו, בדוגמא הזאת יהי בשם record.wav

(3)

```
const int recordLed = 2;
const int mic_pin = A0;
const int sample_rate = 16000;
```

בשורות האלה, ההקלטה מתחילה, האור LED מודלק, והpin של המיקרופון נקבע לA0, קצב המדידה sample\_rate נקבע ל 16000hz.

(4)

```

void wait_min(int mins) {
  int count = 0;
  int secs = mins * 60;
  while (1) {
    Serial.print('.');
    delay(1000);
    count++;
    if (count == secs) {
      count = 0;
      break;
    }
  }
  Serial.println();
  return ;
}

```

בשורות האלה, מתחילים להדפיס למשתמש את הנתונים לפי serial, יודפס למשתמש כל מיני אינפורמציה לגבי זמני ההקלטה.

(5)

```

void setup() {
  // put your setup code here, to run once:
  //initialises the serial connection between the arduino and any connected serial device(e.g.
  computer, phone, raspberry pi...)
  Serial.begin(9600);
  //Sets up the pins
  pinMode(mic_pin, INPUT);
  pinMode(recordLed, OUTPUT);
  Serial.println("loading... SD card");
  if (!SD.begin(SD_CSPin)) {
    Serial.println("An Error has occurred while mounting SD");
  }
  while (!SD.begin(SD_CSPin)) {
    Serial.print(".");
    delay(500);
  }
  audio.CSPin = SD_CSPin;
}

```

עכשיו אנו מתחילים את החיבור בין כרטיס הארדינו למחשב, בקוד לעיל כיווני סיכה מוגדרים וכרטיס ה-SD מותקן. סעיף זה גם נותן לך שגיאה בצג הטורי אם משהו לא בסדר.

(6)

```

void loop() {

Serial.println("#####");
#####");
char fileSINum[20] = "";
itoa(file_number, fileSINum, 10);
char file_name[50] = "";
strcat(file_name, filePrefixname);
strcat(file_name, fileSINum);
strcat(file_name, exten);
Serial.print("New File Name: ");
Serial.println(file_name);
digitalWrite(recordLed, HIGH);
audio.startRecording(file_name, sample_rate, mic_pin);
Serial.println("startRecording ");
// record audio for 2mins. means , in this loop process record 2mins of audio.
// if you need more time duration recording audio then
// pass higher value into the wait_min(int mins) function.
wait_min(2);
digitalWrite(recordLed, LOW);
audio.stopRecording(file_name);
Serial.println("stopRecording");
file_number++;

Serial.println("#####");
#####");
}

```

הקוד לעיל הוא הלולאה האינסופית שבה מקליט הקול מקליט את הקולות. לאחר מכן מאחסן אותם בכרטיס SD שבו מספר הקובץ גדל בכל פעם שהכתיבה בקובץ הושלמה.

ובכך הסתיים הקוד של הארדינו.

MATLAB:

בקוד של Matlab, יש את החלק העיקרי של הפיתרון, כי בעיקר, פה עושים את כל התהליך של זיהוי המקום של קובץ הקול.

(1

קובץ הקול שעובדים עליו ב Matlab הוא בפורמט wav, לכן נשתמש בפונקציה audioread מספרית Matalab שתעביר את הקובץ לתוך שני משתנים כמתואר, y מערך שמייצג Fsi sample data מספר שמייצג sample rate of the data.

```
clear sound;
[file,path] = uigetfile('*.wav');
file_path = append(path,file);
[y,Fs] = audioread(file_path);
```

בקטע קוד זה השתמשנו בפונקציה *uigetfile* שפותחת חלון למשתמש כדי לבחור את הקובץ שהוא צריך להכניס (קובץ הקול), '\*.wav' אומרת שהמשתמש יכול רק לבחור קובץ מפורמט wav.

(2)

```
method = "NCF";
%Method used to estimate pitch, specified as "NCF", "PEF","CEP", "LHS",
or "SRH". The different methods of calculating pitch provide trade-offs in
terms of noise robustness, accuracy, and computation expense. The
algorithms used to calculate pitch are based on the following papers:
% "NCF" — Normalized Correlation Function [1]
range = [70, 800]; % hertz
winDur = 0.08; % seconds
overlapDur = 0.03; % seconds
medFiltLength = 10; % frames
winLength = round(winDur*Fs);
overlapLength = round(overlapDur*Fs);
[F0,loc] = pitch(y,Fs, ...
    'Method',method, ...
    'Range',range, ...
    'WindowLength',winLength, ...
    'OverlapLength',overlapLength, ...
    'MedianFilterLength',medFiltLength);
```

בקטע קוד זה, נבחר את שיטת סינון התדרים מהקובץ, יש כמה שיטות כמתואר בקוד, אבל כדי לקבל כפתרון הכי מדויק, נשתמש ב*NCF method*, יש גם כן את ה*range* של התדרים, ועוד כמה נתונים שפשוט נשים במצב הרגיל שלהם כי לא מעניינות אותנו בפתרון, ואחר כך נשתמש בפונקציית *pitch* שתסנן את התדרים לאורך הקובץ בתקופות זמן כמו שבחרנו *winLength* ותחזיר את התדרים למערך *F0* עם משתנה *loc* שמייצג את אורך המערך.

ב2+1 יש את קטע הקוד שמקבל את קובץ הקול שצריך לעבוד עליו ומסדר אותו בתוך המשתנים והמערכים שנשתמש בהם כדי שנגלה את המקאם, כלומר בקטעי הקוד הבאים יש את הפיתרון העיקרי של אופן גילוי סוג המקאם.

(3)

```

ERTEQAZ = 0;
OCTAVE_ERTEQAZ = 0;
if ERTEQAZ == -1

ERTEQAZ=ERTEQAZ_DETECTOR(new_HZ1,Matching,method,range,winLength,overlapLength,overlapDur,medFiltLength,winDur,Fs1,y1);
    ERTEQAZ
    OCTAVE_ERTEQAZ = floor(ERTEQAZ / 24);
    ERTEQAZ = mod(ERTEQAZ-1,24) + 1;
    OCTAVE_ERTEQAZ
end

```

בקטע קוד זה, כמו שהוסבר קודם, אנו נותנים למשתמש אופציה של בחירת ה-*ERTEQAZ*, בשפה אחרת, הבסיס של *Quranic recitation* שנמצאת בקובץ, אפשריות זאת תיתן למשתמש את האופציה לדיוק מאוד יותר טוב של המקאם, כי כמו שנסתכל בהמשך על אופן גילוי המקאם, נראה שהכל מתבסס על ה-*Erteqaz* (הבסיס). אם הוא בוחר *ERTEQAZ = -1* אז זה אומר שהוא ייתן לנו את קובץ קול מסוים שלפיו נחליט עבור איזה *Erteqaz* הוא הולך להשתמש, אם *erteqaz = 0* אז הוא לא רוצה לבחור, אחרת הוא יבחר מ-*list* שתופיע לו ב-*matlab* את האות שהוא ה-*erteqaz* שלו מבין אותיות C,C#,D,...

(4)

```

global HIJAZ;
global RAST;
global BAYAT;
global NAHAWAND;
global SEKAH;
global AJAM;
global SABA;
global KURD;
global avg;
HIJAZ = [0,2,8,10,14,16,20,24];
RAST = [0,4,7,10,14,18,21,24];
BAYAT = [0,3,6,10,14,16,20,24];
NAHAWAND = [0,4,6,10,14,18,20,24];
SEKAH = [0,3,7,9,14,17,21,24];
AJAM = [0,4,8,10,14,18,22,24];
SABA = [0,3,6,8,14,16,20,24];
KURD = [0,2,6,10,14,16,20,24];

```

בקטע זה, אנו מייצרים שמונה משתנים גלובלים שבכל אחד מהם יש את אופן זיהוי המקאם ביחס ל-*erteqaz*, כלומר כל מערך שייך למקאם אחד שהוא מהווה היחס המרחק בין האותיות של המקאם, היחס הזה הוא יחס בין האותיות לבין ה-*erteqaz*, נשתמש במשתנים האלה אחר כך כדי לעשות *analyzation* של המקאמאת, כלומר חלק נוסף לאחר גילוי המקאם עצמו.

(5) עכשיו נתחיל את הקוד שבו עושים סינון לתדרים שקיבלנו במערך F0 בשלב 2 שהיסברנו קודם, אבל קודם צריך להסביר על כמה משתנים פה,

```

HZmanual =
[32.70,34.65,36.71,38.89,41.20,43.65,46.25,49.00,51.91,55.00,58.27,61.74,65.4
1,69.30,73.42,77.78,82.41,87.31,92.50,98.00,103.83,110.00,116.54,123.47,130.8
1,138.59,146.83,155.56,164.81,174.61,185.00,196.00,207.65,220.00,233.08,246.9
4,261.63,277.18,293.66,311.13,329.63,349.23,
369.99,392.00,415.30,440.00,466.16,
493.88,523.25,554.37,587.33,622.25,659.25,698.46,739.99,783.99,830.61,8
80.00,932.33,987.77];

Matching = zeros(120,1);

BASIC = ERTEQAZ + OCTAVE_ERTQAZ*24;

```

- HZmanual: המערך הזה מייצג את התדר של כל אות מוסיקאלי, התדרים לא משתנים עבור אות יחיד, במערך הזה אנחנו מתחילים מהאות C (ד) באירתיקאז octave 1 ומסיימים ב B (סי) באירתיקאז octave 5.
  - Matching: במערך הזה נתאים כל אות למקום שלו, נאתחל אותו באפסים, הוא באורך 120 כמו ה HZmanual שהוא תחום האותיות שמשתמשים בו.
  - BASIC: הוא נותן לנו את האירתיקאז אם המשתמש בחר אחד, אם לא אז הוא יאותחל לאפס.
- (6) הפונקציה pitch שהתמשנו בה, כאשר היה מתאימה התידרים של קובץ הקול, אם יהיה קטע זמן של שקט, היא תשים את הערך המינמלי בטווח שלה לתוך מערך התדרים, כלומר לפי מה שבחרנו range 70-800 אז אם יהיה לנו שקט בזמן t=9-10 אז נקבל ב F0 בזמן t=9-10 את אוסף מהתדר 70 לאורך זמן זה, לכן נעשה בקטע הקוד הבא את הסינון:

```

l=length(F0);
for i = 1:l
    if F0(i) < 90
        F0(i)= 0;
    end
end

```

- (7) בגלל שאנחנו מדברים על קול של בנאדם Human ולא על כלי נגינה כמו פסנתר או אורגן למשל, אז אנו בטוחים שבאיזשהו קטע זמני מאוד קצר יהיה לבן אדם זה מה שקוראים לו "נשאז", וזה אומר שיש לבן אדם סטייה בתדר שהוא מוציא, בן אדם רגיל לא יכול לזהות את זה דרך שמיעה כי הוא לא מתעניין בדיוק של 0.001 לכן תצריך לזהות במערך F0 את ה"נשאז" הקטן מאוד כדי שיהיה לנו שמירה על רצף של אות אחד עם תדר אחד, גם כן צריך באיזשהו שלב כשעוברים מתדר לתדר אחר, אז יש קטע גם כן מאוד קטן במערך שבו יש קפיצות בין שני תדרים, לכן צריך להפריד בין מקרה של נשאז לבין קפיצה בין שני אותיות, וזה מה שהלולאה הבאה עושה:

```

t = loc/Fs;
l=length(t);
for i = 2:l
    if abs(F0(i)-F0(i-1)) < 2
        F0(i,1)=F0(i-1,1);
    end
end
end

```

(8) במערך HZmanual שהיסברנו עליו קודם, יעבוד רק למקאמאת הלא מזרחית כמו חיגאז – אגאם – נהאואנד – קורד, ואלה מקאמאת שיש בהן יחסים בדיוק של מקסימום חצי טון half tone, אבל בשאר המקאמאת המזרחיות, יש לנו דיוק של רבע טון - quarter tone, לכן כדי לעשות קוד שתומך בכל סוגי המקאמאת, נבנה מערך חדש שיתמוך בquarter tone.

```

new_HZ1 = zeros(120,1);
l= length(HZmanual);

for i=1:l
    k = 2*i - 1;
    new_HZ1(k)=HZmanual(i);
end

l= length(new_HZ1);
for i=1:l
    if new_HZ1(i) ~= 0

    else
        r = log((new_HZ1(i-1))/16.3515978312876);
        r = r/(log(2));
        r = r*1200;
        j = r+50;
        j = j* (log(2));
        j = j/1200;
        j = j+log(16.3515978312876);
        r = exp(j);
        new_HZ1(i)= r;
    end
end
end

```

מה שהקוד הזה עושה, הוא מעתיק בהתחלה את HZmanual למערך חדש בשם new\_HZ1, מעתיק את כל האותיות אבל בין כל שתי אותיות הוא משאיר רווח ביניהם כדי לשים את ה quarter tone, ואז בגלל שיחס התדרים בין האותיות הוא לוגריתמי, אזי משתמשים בשיטה שמתוארת בלולאה השניה שמגלה את ערך התדר החדש אחרי -50 tone כלומר הזזה של התדר ברבע טון.



9) לפי מה שהסברנו קודם בקטע מספר 7 על זיהוי והתקנת תדרים בין נשאז לבין קפיצת אותיות, קטע הקוד הבא מגלה אם יש קפיצה בקוד שלנו ואז הוא ימחק את הקפיצה הזאת כדי שלא תופיע כתופעה של אות מסויים, אלא לקחת את הקצוות של הקפיצה הזאת.

```
diff_f0 = diff(F0(:,1));
l=length(diff_f0);

count = 0;
j=0;
for i=1:l
    if diff_f0(i)~=0
        count = count + 1;
    else
        if count==0
        else
            for k=0:count-2
                F0(k+i-count+1,1)=F0(i-count,1);
            end
        end
        count = 0;
    end
end
end
```

הפונקציה diff מחזירה את ההפרש בין איברים המערך F0 ואז בלולאה בודקים האם יש הפרש בין שני תדרים שלא שווה לאפס, ואז בגלל שאין אופציה לנשאז כי התקנו את התדרים בקטע 7, אזי זה אומר שיש פה קפיצת תדרים, לכן נספור את מספר ההופעות שבהן יש הפרש לא שווה לאפס ואז אחר כך חוזרים על התופעות האלה ומלחיפים את התדרים בתדר של הקצה הראשון, כלומר מוחקים את ההופעות של תדרי קפיצה בין אותיות.

10) בשלב הזה, סיימנו את העבודה של סינון תדרים, כלומר עכשיו יש לנו את מערך התדרים F0 רק עם תדרי אותיות, בלי רעשים כמו נשאז או קפיצות בין אותיות, לכן נשאר לנו רק להתאים את התדרים האלה כל תדר לאות שהוא מציג לפי ה new\_HZ1 manual של frequencies to Musical notes conversion.

לפי מה שהסברנו קודם, בגלל שקול האדם מייצג אותיות שיש בהן נשאז ולא כמו אות שיוצר על ידי כלי נגינה חשמלי, אז תהי לנו סטיות בין התדר שב manual שלנו, לבין התדר עצמו, לכן אופן התאמת האותיות בקטע הקוד הבא הוא אופן שבו מחפשים את התדר מה manual שהכי קרוב לתדר שצריך התאמה וסופרים מספר ההופעות של כל אות המערך Matching.

```

min =1000;
for i = 1:l+1
    for k=1: 120
        if (abs(new_HZ1(k)-F0(i)) < min)
            min = abs(new_HZ1(k)-F0(i,1));
            min_index = k;
        end
    end
    Matching(min_index) = Matching(min_index)+ 1;
    min = 1000;
end

```

(11) בקטע הקוד הבא, נחשב את average ממוצע התדרים בקוד, כלומר סכום הופעות כל אות חלקי מספר האותיות שיש בהן הופעות, נשתמש בavg בהמשך בפונקציות גילוי המקאם.

```

MATCH = zeros(120,1);
sum = 0;
count = 0;
for i = 1:120
    MATCH(i)=Matching(i);
    sum = MATCH(i) + sum;
    if(MATCH(i)>0)
        count = count + 1;
    end
end

avg = sum/count;
avg = avg + Deviation;

```

(12) בקטע הקוד הבא, יהיה לנו שלוש שיטות של Filtering של התדרים כדי שנתאים למקאם:

```

Filtering = 0;
%0 - OFF
%1 - ON
%2 - Advanced Filtering

```

(12.1) אם Filtering == 0 זה אומר שלא נעשה סינון לתדרים אלא ללכת ישר לפונקצית גילוי המקאם, לפי קטע הקוד הבא:

```

if Filtering == 0
    Match2 = zeros(120,1);
    for i=1:120
        if Matching(i)>0
            Match2(i)=1;
        end
    end
    [MAQAM , i] = Maqam_detect(Match2,BASIC);

```

מה שעושים הוא שפשוט נחליף את מספר ההופעות של כל אות ב-1 אם הוא מופיע, ו-0 אם הוא לא מופיע, כדאי להשתמש במצב זה עבור קטע קול שלא מייצג קול של בן אדם אלא רק עבור כלי נגינה חשמליים כי אין להם נשוא אז לא צריך לעשות Filtering.

(12.2) אם  $Filtering == 1$  אזי זה אומר שנעשה סינון תדרים בדרך הכי פשוטה שיש והיא על ידי בדיקה כזאת: אם מספר ההופעות גדול מממוצע ההופעות אז נחליף ב-1 כלומר יש הופעה, אחרת 0, ואז קוראים לפונקציית הגילוי של המקאם. כמתואר בקטע הקוד הבא:

```

elseif Filtering == 1
    for i = 1:120
        if MATCH(i)<avg
            MATCH(i)=0;
        elseif MATCH(i)>avg
            MATCH(i)=1;
        end
    end
    [MAQAM , i] = Maqam_detect(MATCH,BASIC);

```

(12.3) אם  $Filtering == 2$  אזי זה אומר שנעשה סינון תדרים ברמה יותר מתקדמת, בדרך כלל, זה הוא המצב המומלץ עבור Quranic Recitation, לכן נתעניין בפתרון הזה יותר, מה שנעשה בקטע קוד הוא ככה, לפי מה שראינו מהניסויים שלנו עם כל מיני קבצים של Quranic recitations זה שבאופן גילוי של quarter-tone, הקוד מחפש כפול דיוק בתדרי קול, וזה מה שבן אדם לא ממומחה ולא הכי מקצועי שיש לא יכול לעשות, לכן מה שהיה קורה זה שיש שני אותיות שיש הפרש של quarter-tone ביניהם, בשיטה הקודמת נותנת שיש בשני האותיות האלה הופעות של קול, אבל זה לא באמת נכון, זה מה שקוראים לו נשוא, לא אותו נשוא כמו. בשלבים קודמים אלא זה נשוא שמיוצג על ידי סטיית התאמה במערכת זיהוי והתאמת תדרי הקול לאותיות, לכן האלגוריתם הבא יתקן את כל הבעיה הזאת, הדרך היא שנחפש במערך התדרים על שתי הופעות שיש ביניהם quarter-tone ושתיהן מעל averagen ואז למחוק אחת מהן, רעיון המחיקה הוא שנמחק את התו מוזיקלי שפחות מופיע מהתו מוזיקלי השני, אבל גם כן זה יגרום לעוד בעיה, שאם יש שני אותיות שיש ביניהם half-tone אז לתו מוזיקלי שביניהם שהוא עם quarter-tone on אחד, יישאר לנו רק הופעה של תדר אחד מבין שלושת התדרים, לכן גם כן. במצב הזה נמחק את התדר הבינוני, כלומר התו מוזיקלי שנמצא באמצע בין שני אותיות, ואז נקרא לפונקציית זיהוי המקאם.

```

[MAQAM , i] = Maqam_detect(Match2,BASIC);
elseif Filtering == 2
    Match = Normal_Filter(Matching,Deviation);
    diff_Filter = diff(Match);
    for i=1:119
        if diff_Filter(i)~=0 || (Match(i+1)>avg && Match(i)>avg)
            if Match(i+1)>Match(i)
                Match(i)=0;
            else
                Match(i+1)=0;
            end
        end
    end
    Match3 = zeros(120,1);
    for i=1:120
        if Match(i)>0
            Match3(i)=1;
        end
    end
    [MAQAM , i] = Maqam_detect(Match3,BASIC);
end

```

13) פונקציית זיהוי המקאם היא פונקציה הכי פשוטה שיש, היא רק מחפשת יחס בין אותיות שמקיים את אותו יחס של איזשהו מקאם, לפי המשתנים הגלובליים שהגדרנו קודם, המקאם הראשון שהיא תמצא, הוא יהיה המקאם המוחזר כתשובה, הקטע מחולק לשני חלקים, אם 0==BASIC כלומר אם המשתמש לא מתן לנו את האירטקאז שהוא רוצה, אזי נעבוד בצורה רנדומלית של חיפוש מקאם, כך שהתו מוזיקלי הראשון שנמצא לו הופעה, נסתמך עליו כהאירטקאז ואז נתחיל לחפש את המקאם לפיו, אחרת אם המשתמש נתן לנו את האירטקאז שהוא רוצה כדי שיהיה לו יותר דיוק, אזי מתחילים לחפש החל ממקום האירטקאז שהוא נתן, אם נמצא מאקאם שמתאים לאירטקאז אז נחזיר, אחרת בשני המקרים אם לא מצאנו מקאם שמתאים עבור יחסי האותיות של המקאמאת, אזי נחזיר .ERROR

```

function [MAQAM,i] = Maqam_detect(Matching,BASIC)
% find first 1 in matching array
length = 120;
%
if BASIC == 0
    for i=1:length
        if (Matching(i)==1)
            % find second 1 in matching array
            for k=i+1:length
                if Matching(k)==1
                    % determine what is the distance between two 1 to
                    % the maqams to better analysation
                    % if the distance is 1 so we will search for kurd
                    % hijaz
                    % if the distance is 2 so we will search for
                    % or ajam
                    if k == i+2
                        for s=k+1:length
                            % find 3rd 1
                            if Matching(s) == 1
                                if s == k+4
                                    % distance is 1-2 so KURD
                                    MAQAM = "KURD";
                                    return
                                elseif s == k+6
                                    % distance is 1-3 so HIJAZ
                                    MAQAM = "HIJAZ";
                                    return
                                end
                            end
                        end
                    elseif k == i+4
                        for s=k+1:length
                            if Matching(s) == 1
                                if s== k+2
                                    % distance is 2-1 so NAHAWAND
                                    MAQAM = "NAHAWAND";
                                    return
                                elseif s== k+4
                                    % distance is 2-2 so AJAM
                                    MAQAM = "AJAM";
                                    return
                                elseif s == k+3
                                    MAQAM = "RAST";
                                    return
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

end
elseif k == i+3
    for s=k+1:length
        if Matching(s) == 1
            if s == k+4
                MAQAM = "SEKAH";
                return
            elseif s == k+3
                for n=s+1:length
                    if Matching(n) == 1
                        if n == s+4
                            MAQAM = "BAYAT";
                            return
                        elseif n == s+2
                            MAQAM = "SABA";
                            return
                        end
                    end
                end
            end
        end
    end
end
end
end
end
end
end
end
end
elseif BASIC > 0
    i=BASIC;
    if (Matching(i)==1)
        % find second 1 in matching array
        for k=i+1:length
            if Matching(k)==1
                % determine what is the distance between two 1 to
split
                % the maqams to better analysation
or
                % if the distance is 1 so we will search for kurd
                % hijaz
                % if the distance is 2 so we will search for
nahawand
                % or ajam
                if k == i+2
                    for s=k+1:length
                        % find 3rd 1
                        if Matching(s) == 1
                            if s == k+4
                                % distance is 1-2 so KURD

```

```

        MAQAM = "KURD";
        return
    elseif s == k+6
        % distance is 1-3 so HIJAZ
        MAQAM = "HIJAZ";
        return
    end
end
end
elseif k == i+4
    for s=k+1:length
        if Matching(s) == 1
            if s== k+2
                % distance is 2-1 so NAHAWAND
                MAQAM = "NAHAWAND";
                return
            elseif s== k+4
                % distance is 2-2 so AJAM
                MAQAM = "AJAM";
                return
            elseif s == k+3
                MAQAM = "RAST";
                return
            end
        end
    end
end
elseif k == i+3
    for s=k+1:length
        if Matching(s) == 1
            if s == k+4
                MAQAM = "SEKAH";
                return
            elseif s == k+3
                for n=s+1:length
                    if Matching(n) == 1
                        if n == s+4
                            MAQAM = "BAYAT";
                            return
                        elseif n == s+2
                            MAQAM = "SABA";
                            return
                        end
                    end
                end
            end
        end
    end
end
end
end
end

```

```

        end
    end
end
MAQAM="ERROR";
return
end

```

(14) לפי מה שהסברנו בשלב 3, יש לנו פה את הפונקציה שתמצא את האירטקאז:

```

function [ERTEQAZ] =
ERTEQAZ_DETECTOR(new_HZ1,Matching,method,range,winLength,overlapLength,ove
rlapDur,medFiltLength,winDur,Fs1,y1) %#ok<INUSL>
    winLength = round(winDur*Fs1);
    overlapLength = round(overlapDur*Fs1);
    [F1,~] = pitch(y1,Fs1, ...
        'Method',method, ...
        'Range',range, ...
        'WindowLength',winLength, ...
        'OverlapLength',overlapLength, ...
        'MedianFilterLength",medFiltLength);

    l=length(F1);

    min =1000;
    for i = 1:l
        for k=1: 120
            if (abs(new_HZ1(k)-F1(i)) < min)
                min = abs(new_HZ1(k)-F1(i,1));
                min_index = k;
            end
        end
        Matching(min_index) = Matching(min_index)+ 1;
        min = 1000;
    end
    Matching(1)=0;
    max = 0;
    ERTEQAZ = 0;
    for i=1:120
        if(Matching(i)>max)
            max =Matching(i);
            ERTEQAZ = i;
        end
    end
    return
end

```



היא עובדת באותו אופן של Filtering = 0 כלומר היא מצפה שהמשתמש ייתן את הקובץ שמייצג את האיריתקאז כתדר בלי נשאז וכתדר שלא צריך שום סוג של סינון, אלא תו מוזיקלי מכלי נגינה חשמלי, והיא מחפשת במערך את ההופעה של התו מוזיקלי הזה, ואז זה יישמר כהאיריתקאז של .recitation

(15) קטע הקוד הבא מייצג את פונקציית Normal\_Filtering שעושה סינון תדרים באופן שהוסבר בסעיף 12.3, כלומר בודקת האם ההופעה של תו מוזיקלי מסוים היא יותר גדולה מהממוצע.

```
function [MATCH] = Normal_Filter(Match,Deviation)
    sum=0;
    count=0;
    MATCH = zeros(120,1);
    for i = 1:120
        MATCH(i)=Match(i);
        sum = MATCH(i) + sum;
        if(MATCH(i)>0)
            count = count + 1;
        end
    end

    avg = sum/count;
    avg = avg + Deviation - 1;
    for i = 1:120
        if MATCH(i)<avg
            MATCH(i)=0;
        end
    end
    return
end
```

(16) עכשיו לאחר שסיימנו את תהליך זיהוי המקאם, נרצה לתת למשתמש עוד אינפורמציה לגבי כמה פרטים, נשאל המשתמש אם הוא ירצה ניתוח מפורש על:

```
analysisation = true;
```

analysisation==true אומרת שהמשתמש רוצה לצפות בניתוח, אחרת false.

(17) אם analysisation == true נריץ קטע קוד הבא:

```
if analysisation
    MAQAM_ANALYSATION(Matching,MAQAM,i);
end
```

(17.1) נכניס את המשתנים הגלובליים שיצרנו בהתחלה כדי שנשתמש בהם, אחר כך נסתכל על המקאם שקיבלנו, נתאים ערך עבור כל מקאם כמתואר בקוד, למשל אם המקאם הוא

Ajam אז  $k=2$ , ואז כדי שנתחיל הניתוח נספור את מספר ההופעות הכללי של כל תו מוזיקלי במערך Matching ונשים את הערך במשתנה count שנשתמש בו אחר כך.

```
function [] = MAQAM_ANALYSATION(Matching,MAQAM,ER)
    global HIJAZ;
    global RAST;
    global BAYAT; %#ok<*GVMIS>
    global NAHAWAND;
    global SEKAH;
    global AJAM;
    global SABA;
    global KURD;
    global avg;
    count = 0;
    if MAQAM == "RAST"
        k=1;
    elseif MAQAM == "AJAM"
        k=2;
    elseif MAQAM == "NAHAWAND"
        k=3;
    elseif MAQAM == "BAYAT"
        k=4;
    elseif MAQAM == "KURD"
        k=5;
    elseif MAQAM == "HIJAZ"
        k=6;
    elseif MAQAM == "SABA"
        k=7;
    elseif MAQAM == "SEKAH"
        k=8;
    end
    for i=1: length(Matching)
        count = count + Matching(i);
    end
```

17.2) יש לנו את הפונקציה שתעשה כל מיני סוגי ניתוח Matching\_Rate נראה אותה בהמשך, הפונקציה מבקשת את המקאם שתעשה ניתוח עליו, גם כן את האירתיקאז שלו, עם המשתנה count שחישבנו קודם והמערך Matching, בתנאים שבקוד, אנו בודקים מה הוא המקאם שקיבלנו, אם הוא Rast או Ajam או Nahawand, אזי בגלל שלמקאמאט - Maqams האלה יש אותו האירתיקאז יחסית, כלומר המרחק בתווים מוזיקליים ביניהם לבין שאר המקאמאט קבוע, לכן נעשה קריאה לפונקציה Matching\_Rate באותו תנאי if, נקרא 8 קריאות לפונקציה Matching\_Rate קריאה עבור כל מקאם בשלושה חלקים: את Rast Ajami Nahawandi עם אותו אירתיקאז, את Bayat Kurdi Hijazi Sabai גם כן עם אותו אירתיקאז פלוס המרחק בתווים מוזיקליים שהוא \*quarter-tone4, ובסוף את Sekah עם אותו אירתיקאז פלוס המרחק שהוא \*quarter-tone7.

באותו אופן נבצע שלב זה עם שאר המקאמאט Bayat Hijazi Kurdi Sabai עם אותו אירתיקאז, והה Sekah לבד.

```

if MAQAM == "RAST" || MAQAM == "AJAM" || MAQAM == "NAHAWAND"
    ra_pr = Matching_Rate(ER,"RAST",count,Matching);
    aj_pr = Matching_Rate(ER,"AJAM",count,Matching);
    na_pr = Matching_Rate(ER,"NAHAWAND",count,Matching);
    by_pr = Matching_Rate(ER+4,"BAYAT",count,Matching);
    ku_pr = Matching_Rate(ER+4,"KURD",count,Matching);
    hi_pr = Matching_Rate(ER+4,"HIJAZ",count,Matching);
    sa_pr = Matching_Rate(ER+4,"SABA",count,Matching);
    se_pr = Matching_Rate(ER+7,"SEKAH",count,Matching);
elseif MAQAM == "BAYAT" || MAQAM == "KURD" || MAQAM == "HIJAZ" ||
MAQAM == "SABA"
    ra_pr = Matching_Rate(ER-4,"RAST",count,Matching);
    aj_pr = Matching_Rate(ER-4,"AJAM",count,Matching);
    na_pr = Matching_Rate(ER-4,"NAHAWAND",count,Matching);
    by_pr = Matching_Rate(ER,"BAYAT",count,Matching);
    ku_pr = Matching_Rate(ER,"KURD",count,Matching);
    hi_pr = Matching_Rate(ER,"HIJAZ",count,Matching);
    sa_pr = Matching_Rate(ER,"SABA",count,Matching);
    se_pr = Matching_Rate(ER+3,"SEKAH",count,Matching);
elseif MAQAM == "SEKAH"
    ra_pr = Matching_Rate(ER-7,"RAST",count,Matching);
    aj_pr = Matching_Rate(ER-3,"AJAM",count,Matching);
    na_pr = Matching_Rate(ER-3,"NAHAWAND",count,Matching);
    by_pr = Matching_Rate(ER-3,"BAYAT",count,Matching);
    ku_pr = Matching_Rate(ER-3,"KURD",count,Matching);
    hi_pr = Matching_Rate(ER-3,"HIJAZ",count,Matching);
    sa_pr = Matching_Rate(ER-3,"SABA",count,Matching);
    se_pr = Matching_Rate(ER,"SEKAH",count,Matching);
else
    return
end

```

17.3) הפונקציה Matching\_Rate בודקת את אחוז ההתאמה בין כל מקאם לבין מספר אחוז ההופעות לבין התווים המוזיקליים לכל מקאם, כלומר מה שעושים הוא שמחלקים את סכום מספר בהופעות של התווים המוזיקליים ביחס לאירתקאז הנתון חלקי מספר ההופעות של התווים המוזיקליים בכללי, וזה מהווה לנו את אחוז המקאם בקטע קול המצורף, לכן עבור על מקאם אנו מחשבים סכום מספר ההופעות של התווים שלו, התווים האלה יש ביניהם יחס קבוע (מרחק קבוע) שמחשבים לפי האירתקאז, לכן אחרי שמחשבים את היחס הזה אנו מחזירים את אחוז הופעת המקאם הזה בקובץ שעושים ניתוח עליו.

```

function [percent] = Matching_Rate(ER,MAQAMM,count,Matching)
    keys = 0;
    i = ER;
    if MAQAMM == "RAST"
        keys = keys + Matching(i);
        keys = keys + Matching(i+4);
        keys = keys + Matching(i+7);
        keys = keys + Matching(i+10);
        keys = keys + Matching(i+14);
        keys = keys + Matching(i+18);
        keys = keys + Matching(i+21);
        keys = keys + Matching(i+24);
        percent = keys/count * 100;
        return
    elseif MAQAMM == "AJAM"
        keys = keys + Matching(i);
        keys = keys + Matching(i+4);
        keys = keys + Matching(i+8);
        keys = keys + Matching(i+10);
        keys = keys + Matching(i+14);
        keys = keys + Matching(i+18);
        keys = keys + Matching(i+22);
        keys = keys + Matching(i+24);
        percent = keys/count * 100;
        return
    elseif MAQAMM == "NAHAWAND"
        keys = keys + Matching(i);
        keys = keys + Matching(i+4);
        keys = keys + Matching(i+6);
        keys = keys + Matching(i+10);
        keys = keys + Matching(i+14);
        keys = keys + Matching(i+18);
        keys = keys + Matching(i+20);
        keys = keys + Matching(i+24);
        percent = keys/count * 100;
        return
    elseif MAQAMM == "BAYAT"
        keys = keys + Matching(i);
        keys = keys + Matching(i+3);
        keys = keys + Matching(i+6);
        keys = keys + Matching(i+10);
        keys = keys + Matching(i+14);
        keys = keys + Matching(i+18);
        keys = keys + Matching(i+21);
        keys = keys + Matching(i+24);
        percent = keys/count * 100;
        return

```

```

elseif MAQAMM == "KURD"
    keys = keys + Matching(i);
    keys = keys + Matching(i+2);
    keys = keys + Matching(i+6);
    keys = keys + Matching(i+10);
    keys = keys + Matching(i+14);
    keys = keys + Matching(i+16);
    keys = keys + Matching(i+20);
    keys = keys + Matching(i+24);
    percent = keys/count * 100;
    return
elseif MAQAMM == "HIJAZ"
    keys = keys + Matching(i);
    keys = keys + Matching(i+2);
    keys = keys + Matching(i+8);
    keys = keys + Matching(i+10);
    keys = keys + Matching(i+14);
    keys = keys + Matching(i+16);
    keys = keys + Matching(i+20);
    keys = keys + Matching(i+24);
    percent = keys/count * 100;
    return
elseif MAQAMM == "SABA"
    keys = keys + Matching(i);
    keys = keys + Matching(i+3);
    keys = keys + Matching(i+6);
    keys = keys + Matching(i+8);
    keys = keys + Matching(i+14);
    keys = keys + Matching(i+16);
    keys = keys + Matching(i+20);
    keys = keys + Matching(i+24);
    percent = keys/count * 100;
    return
elseif MAQAMM == "SEKAH"
    keys = keys + Matching(i);
    keys = keys + Matching(i+3);
    keys = keys + Matching(i+7);
    keys = keys + Matching(i+9);
    keys = keys + Matching(i+15);
    keys = keys + Matching(i+17);
    keys = keys + Matching(i+21);
    keys = keys + Matching(i+24);
    percent = keys/count * 100;
    return
end
end

```

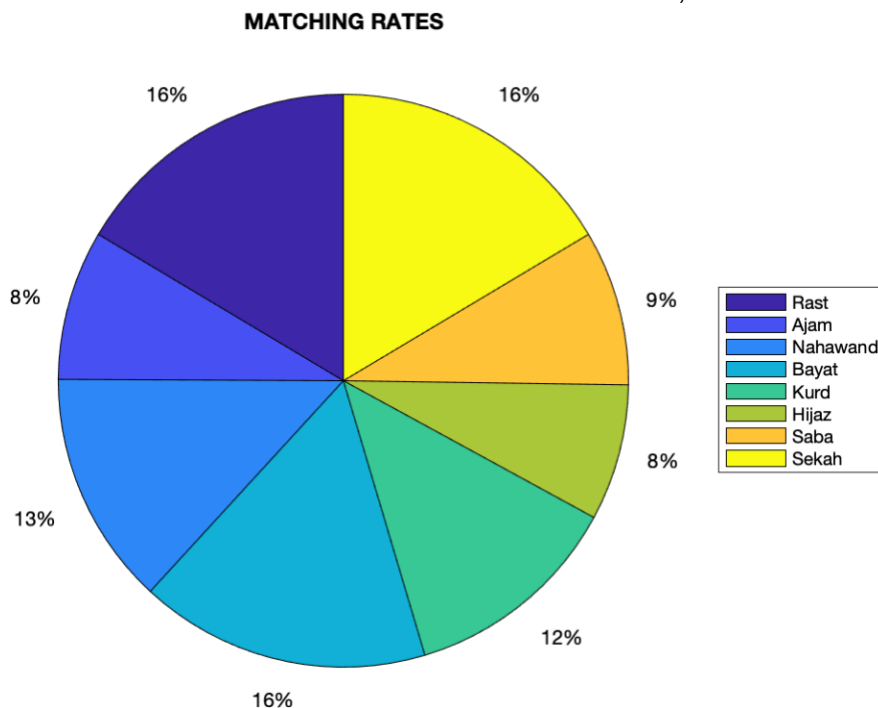
17.4) בקטע קוד זה, אנו מדפיסים את האחוזים שקיבלנו מ17.2 לגבי כל מקאם.

```
fprintf("MATCHING RATES ARE:\n");
fprintf("Rast Matching Rate = %f \n", ra_pr);
fprintf("Ajam Matching Rate = %f \n", aj_pr);
fprintf("Nahawand Matching Rate = %f \n", na_pr);
fprintf("Bayat Matching Rate = %f \n", by_pr);
fprintf("Kurd Matching Rate = %f \n", ku_pr);
fprintf("Hijaz Matching Rate = %f \n", hi_pr);
fprintf("Saba Matching Rate = %f \n", sa_pr);
fprintf("Sekah Matching Rate = %f",se_pr);
```

17.5) אחרי סיום השלב הראשון של הניתוח אנו מתחילים להדפיס התוצאות שקיבלנו ממצב זה, בשני השלבים האלה אנו מייצגים שני תרשימים, התרשים הראשון הוא מסוג pie, בקוד, אנו נותנים לפונקציית התרשים את הערכים של האחוזים, בנוסף לשמות עבור כל צבע ב pie כמתואר:

```
f1 = figure;
y2011 = [ra_pr,aj_pr,na_pr,by_pr,ku_pr,hi_pr,sa_pr,se_pr];
labels =
{'Rast','Ajam','Nahawand','Bayat','Kurd','Hijaz','Saba','Sekah'};
ax2 = nexttile;
pie(ax2,y2011);
title('MATCHING RATES');
lgd = legend(labels);
lgd.Layout.Tile = 'east';
```

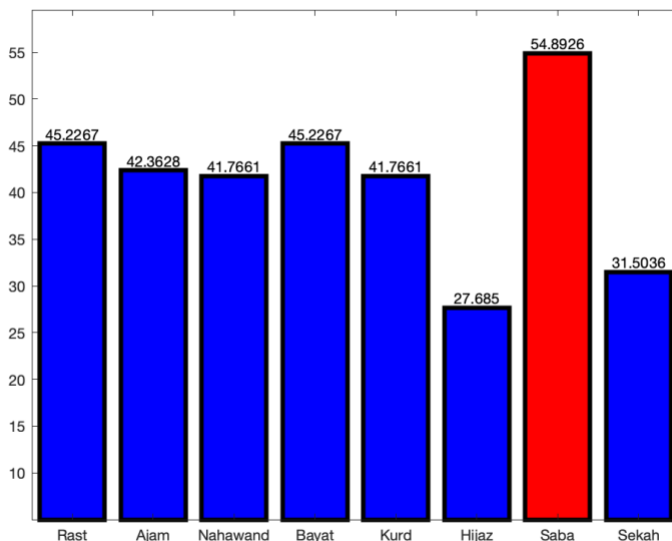
ואז אנו מקבלים למשל עבור דוגמה מסוימת, את התרשים הבא:



17.6) התרשים השני שאנו מייצגים למשתמש לפי הניתוח שנערך, הוא תרשים מסוג bar, בקוד, אנו נותנים לפונקציה שמייצרת את התרשים את הנתונים המבוקשים, ה bar יהיה מורכב מעמודים, כך שכל עמוד מייצג את אחוז הופעת כל מקאם לבד, ואנו צובעים את העמוד של המקאם שקיבלנו בצבע אדום כדי שיהיה יותר קל לנתח את התרשים ולעשות כל מיני השוואות. הקוד הוא:

```
f2 = figure;
X =
categorical({'Rast','Ajam','Nahawand','Bayat','Kurd','Hijaz','Saba','Sekah'
});
X =
reordercats(X,{'Rast','Ajam','Nahawand','Bayat','Kurd','Hijaz','Saba','Sekah'
});
Y = [ra_pr ; aj_pr ; na_pr ; by_pr ; ku_pr ; hi_pr ; sa_pr ; se_pr];
b = bar(X,Y,'FaceColor',[0 0 1],'EdgeColor',[0 0 0],'LineWidth',3);
b.FaceColor = 'flat';
b.CData(k,:) = [1 0 0];
xtips2 = b(1).XEndPoints;
ytips2 = b(1).YEndPoints;
labels2 = string(b(1).YData);
text(xtips2,ytips2,labels2,'HorizontalAlignment','center',...
'VerticalAlignment','bottom')
```

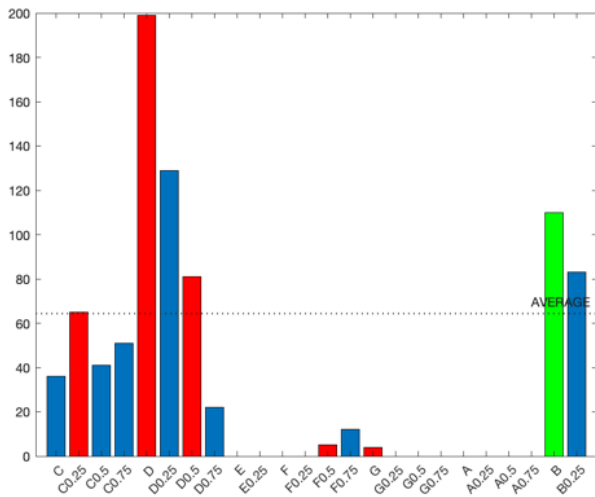
ואז אנו מקבלים למשל עבור דוגמה מסוימת, את התרשים הבא:



17.7) בשלב הזה, נדפיס למשתמש תקשים שמייצג את מספר ההופעות של כל תוו מוזיקלי, התרשים יהיה מסוג bar שכל עמוד מייצג תוו מוזיקלי בדיוק של quarter-tone בין התווים, בנוסף שהתווים עבור המקאם יהיו צבועים בצבע אדום והאירתקאז צבוע בצבע ירוק, גם כ' יהיה קו מקווקו המקביל לציר אשמייצג את הממוצע של ההופעות שחישבנו קודם בשלבי סינון התדרים, הקוד הוא:

```
f3 = figure;
X =
categorical({'C','C0.25','C0.5','C0.75','D','D0.25','D0.5','D0.75','E','E0.25',
'F','F0.25','F0.5','F0.75','G','G0.25','G0.5','G0.75','A','A0.25','A0.5',
'A0.75','B','B0.25'});
X =
reordercats(X,{'C','C0.25','C0.5','C0.75','D','D0.25','D0.5','D0.75','E','E0.25',
'F','F0.25','F0.5','F0.75','G','G0.25','G0.5','G0.75','A','A0.25','A0.5',
'A0.75','B','B0.25'});
Notes = zeros(24,1);
for i=1:120
    k = mod(i-1,24) + 1;
    Notes(k) = Notes(k) + Matching(i);
end
bb = bar(X,Notes,'stacked');
bb.FaceColor = 'flat';
yligne(avg,':','AVERAGE','LineWidth',1.5);
```

התרשים עבור דוגמה כלשהי, יהיה באופן הבא:



17.8) בשלב האחרון של כל תהליך הניתוח, נדפיס למשתמש את מספר ההופעות עבור כל תוו מוזיקלי בכל אוקטבה – Octave, מהאוקטבה הראשונה עד החמישית, ההצגה תהיה באופן הבא:

כל תרשים מייצג אוקטבה בצורת bar שכל עמוד מייצג את מספר ההופעות של כל התווים המוזיקליים בדיוק של quarter-tone.



הקוד הוא:

```

ER = mod(ER-1,24) + 1;
if MAQAM == "RAST"
    for i=1:7
        l = RAST(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
elseif MAQAM == "AJAM"
    for i=1:7
        l = AJAM(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
elseif MAQAM == "NAHAWAND"
    for i=1:8
        l = NAHAWAND(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
elseif MAQAM == "BAYAT"
    for i=1:7
        l = BAYAT(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
end

```

```

elseif MAQAM == "KURD"
    for i=1:7
        l = KURD(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
elseif MAQAM == "HIJAZ"
    for i=1:7
        l = HIJAZ(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
elseif MAQAM == "SABA"
    for i=1:7
        l = SABA(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
elseif MAQAM == "SEKAH"
    for i=1:7
        l = SEKAH(i) + ER;
        l = mod(l-1,24) + 1;
        if i == 1
            bb.CData(l,:) = [0 1 0];
        else
            bb.CData(l,:) = [1 0 0];
        end
    end
end
end

```

הקוד. הזה ייחשב עבור כל אוקטבה - Octave את מספר ההופעות של כל תו מוזיקלי, ואחר כך

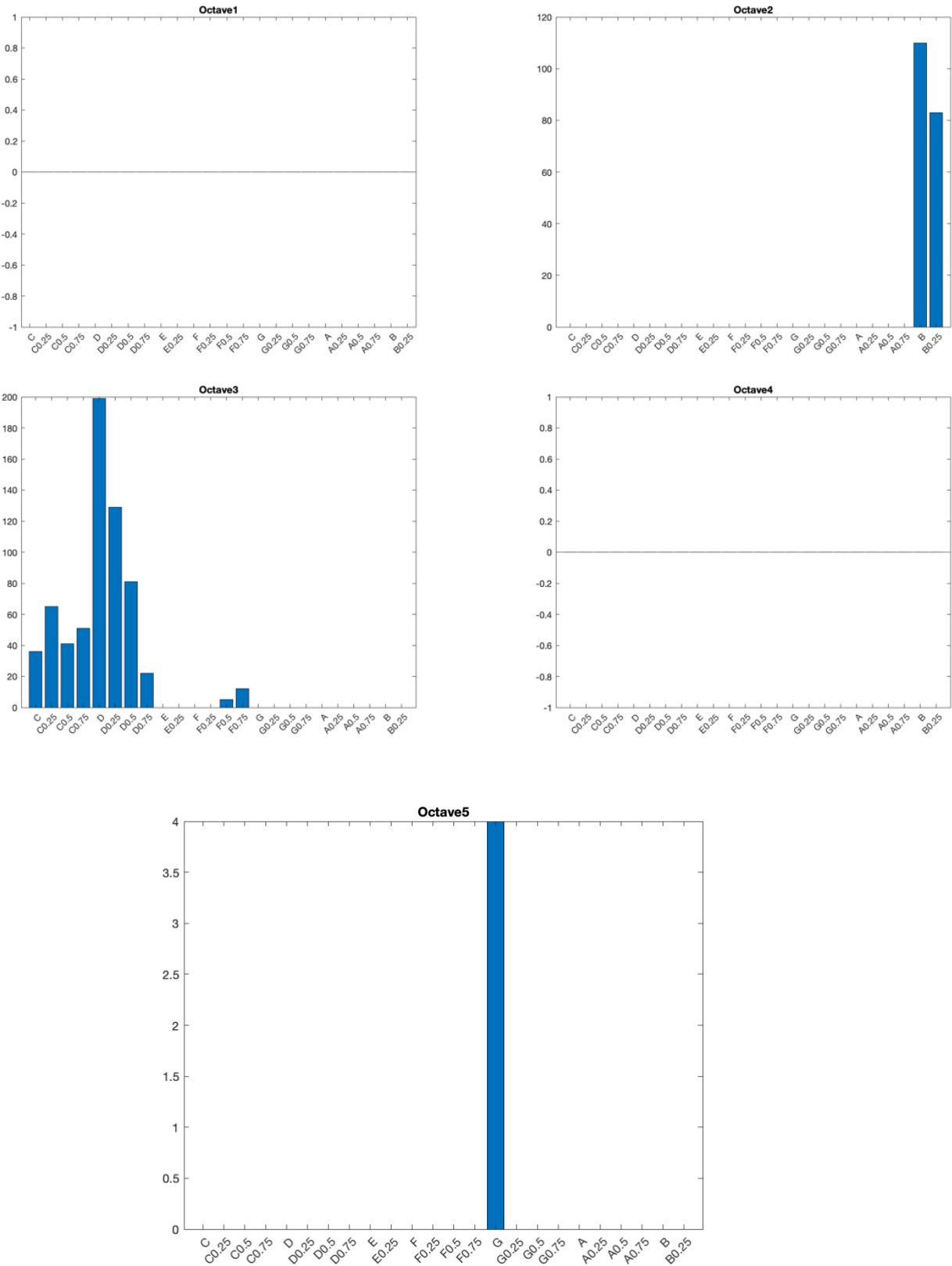
בקוד הבא אנו נדפיס את התרשימים עבור כל אוקטבה.

```

Octave = zeros(24,1);
for i=1:120
    k = mod(i-1,24) + 1;
    Octave(k) = Octave(k) + Matching(i);
    if (mod(i,24)==0)
        figure;
        nexttile;
        X =
categorical({'C','C0.25','C0.5','C0.75','D','D0.25','D0.5','D0.75','E','E0
.25','F','F0.25','F0.5','F0.75','G','G0.25','G0.5','G0.75','A','A0.25','A0
.5','A0.75','B','B0.25'});
        X =
reordercats(X,{'C','C0.25','C0.5','C0.75','D','D0.25','D0.5','D0.75','E','
E0.25','F','F0.25','F0.5','F0.75','G','G0.25','G0.5','G0.75','A','A0.25','
A0.5','A0.75','B','B0.25'});
        bar(X,Octave,'stacked');
        chr = int2str(floor((i-1)/24) + 1);
        tt = "Octave" + chr;
        title(tt);
        Octave = zeros(24,1);
    end
end

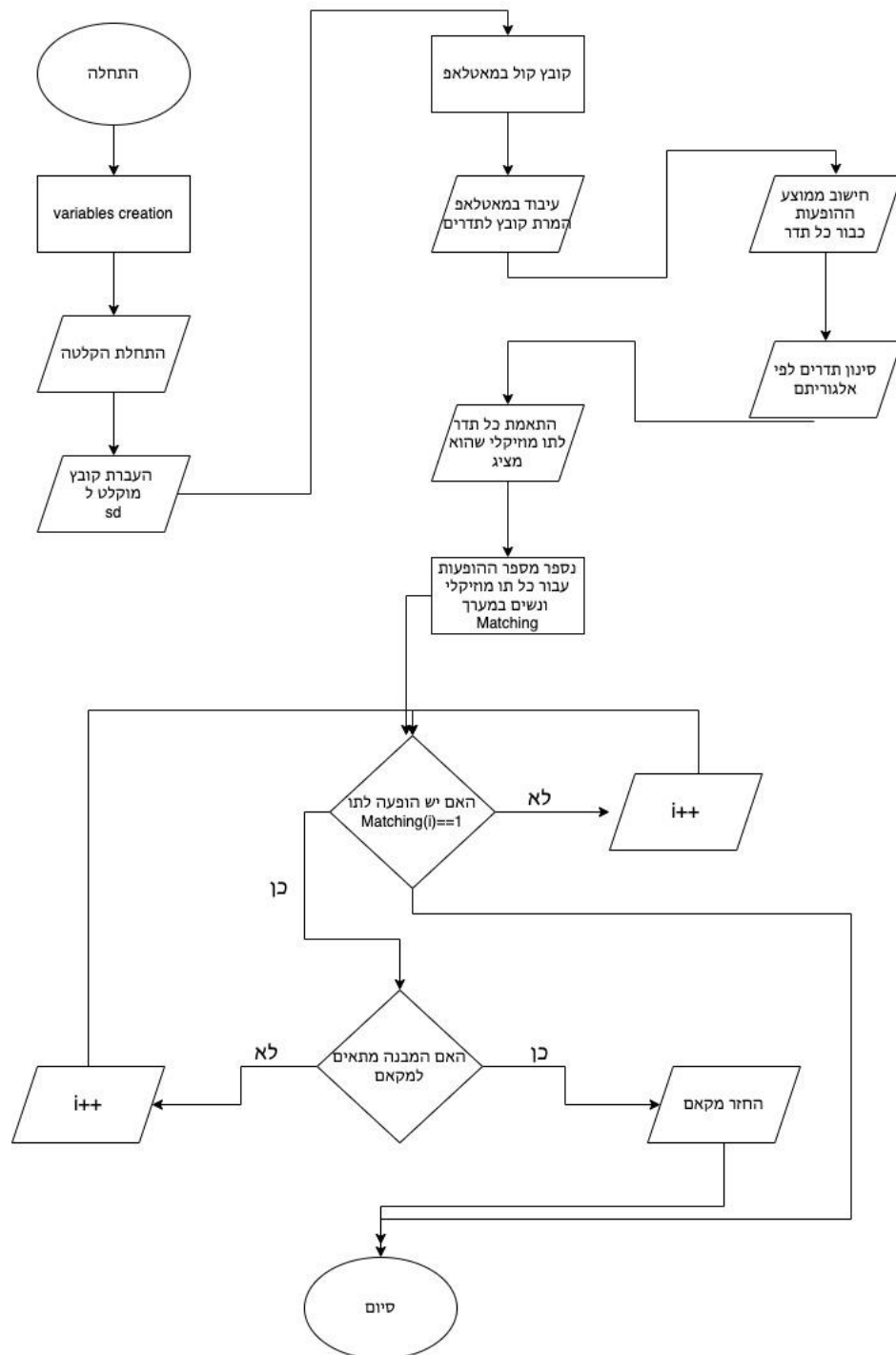
```

והתוצאה עבור דוגמה כלשהי:



ובכך הסתיים הקוד.

5.2.4 תוכנה סופית +הסברים כללים.



## ביבליוגרפיה

- [illegible]

