

# בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

מבוא למדעי המחשב 67101

## תרגיל 6 – עריכת קבצי אודיו

להגשה בתאריך 11/12/2019 בשעה 22:00

### הקדמה

בתרגיל זה נכתוב תכנית לעריכת קבצי אודיו מסוג wav. לצורך פתרון התרגיל עליכם להוריד את הקובץ `wave_helper.py` – מודול זה כבר מומש בשבילכם, והוא מכיל מספר פונקציות הדרושות לתרגיל. אל תעשו שום שינוי בקובץ זה! מטרת התרגיל היא כתיבת תכנית המקבלת מהמשתמש קבצי wav, משנה את הקבצים לפי הנחיות המשתמש ושומרת את הגרסה החדשה של הקבצים כקובץ חדש. התוכנית מאפשרת לשנות שמע בקבצים קיימים ולהלחין מנגינה בקובץ חדש. צריך להגיש את התרגיל בזוגות. הקפידו לעקוב אחר ההנחיות המפורטות בהוראות ההגשה בסוף התרגיל.

### הספריות numpy ו-scipy

הקובץ `wave_helper.py` – הנתון לכם בתרגיל זה עושה שימוש בספריות `numpy` ו-`scipy` של פייתון, ולכן נצטרך אותן בשביל להריץ את התרגיל. במעבדת המחשבים של האוניברסיטה (האקווריום) כבר מותקנות ספריות אלו ואין צורך להתקין שום דבר. כמו כן, הספריות כלולות ב- **WinPython** אותו הורדתם בתחילת הסמסטר. במידה ואתם עובדים ב-`virtualenv` בתוך `pycharm` או על `Mac`, התקינו את הספריות ע"י הרצת הפקודות הבאות מהטרמינל שלכם:

```
pip install scipy
pip install numpy
```

### דגשים לתרגיל

- אנו ממליצים לכתוב את כל התוכנית בקובץ אחד ושמו `wave_editor.py`. את החלוקה לפונקציות עליכם לבצע על פי שיקול דעתכם, לפי העקרונות שנלמדו בקורס. הדגש צריך להיות על קוד מודולרי (ללא כפל קוד), ברור וקריא. את החלוקה לפונקציות יש לנמק בקובץ נוסף ששמו יהיה `description.txt`. ניתן להוסיף קבצים נוספים אך יש לנמק את הבחירה בקובץ `description.txt` ולוודא שההרצה עובדת.
- כל הקבצים יוגשו בתוך קובץ `ex6.zip`. ניתן להניח שכל הקבצים המוגשים בקובץ זה יימצאו בתיקיה בעת הקריאה לסקריפט.

# בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- סגנון: הקפידו על תיעוד נאות ובחרו שמות משתנים משמעותיים. הקפידו להשתמש בקבועים (שמות משתנים באותיות גדולות) על פי הצורך.

## תיאור התרגיל

תיאור התרגיל מורכב משני חלקים. החלק הראשון מתאר את אופן פעולת התוכנית - כיצד התוכנית שלכם צריכה לעבוד. החלק השני של תיאור התרגיל מספק את הרקע הנדרש לעבודה עם קבצי wav, וכן מפרט כיצד צריך להשתמש בפקודות בקובץ העזר wave\_helper.py. אנו ממליצים לקרוא את שני החלקים לפני שתתחילו לכתוב את הקוד.

## תיאור כללי של התכנית

עבור כל אחד מהתפריטים המתוארים בעברית, תדרשו לכתוב גרסה מתאימה באנגלית שתוצג למשתמש.

סדר הפעולות אותו אנו מציינים צריך להישמר, אך מעבר לכך הניסוח נתון לשיקול דעתכם 😊

בעת הרצת התכנית יוצג למשתמש תפריט הכניסה, המציג בפניו את הפעולות אותן התוכנה מאפשרת לו לעשות :

1. שינוי קובץ wav.

2. הלחנת מנגינה בפורמט המתאים לקובץ wav.

3. יציאה מהתכנית.

בשביל לבחור את אחת הפעולות, על המשתמש לכתוב את מספר הפקודה אותה ירצה לבצע ולהקיש על Enter.

### 1. תפריט שינוי קובץ wav (אופציה 1)

במידה והמשתמש בוחר לשנות קובץ wav, תוצג למשתמש הודעה המבקשת שיכתוב את שם הקובץ אותו ירצה לקרוא.

לאחר שהמשתמש יכניס את שם הקובץ, יוצג בפניו תפריט שינוי הקבצים, המציג בפניו המשתמש את ששת השינויים אותם התוכנה מאפשרת לו לעשות על השמע :

1. היפוך

2. האצת מהירות

3. האטת מהירות

4. הגברת ווליום

5. הנמכת ווליום

6. פילטר עימעום (low pass filter)

7. מעבר לתפריט הסיום

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

לאחר שהמשתמש בוחר באחת האופציות, התוכנית מבצעת את השינוי שהמשתמש בחר על השמע, כותבת הודעה שהשינוי הרלוונטי התבצע, וחוזרת לתפריט שינוי קובץ wav כאשר השמע עליו תפעל כעת הוא השמע ששינונו. אם המשתמש בוחר לעבור לתפריט הסיום מבלי לבצע פעולות נוספות על השמע, השמע נשמר עם השינויים שביצענו בו עד כה. אם לא ביצענו שינויים בכלל, השמע ישמר כפי שהתקבל.

### 2. הלחנת מנגינה (אופציה 2)

במידה והמשתמש בוחר להלחין מנגינה, התוכנה תציג למשתמש הודעה בה היא מבקשת את שם קובץ הנחיות ההלחנה. התוכנה תצפה לקבל קובץ בפורמט המפורט בפרק "הלחנת מנגינה". במידה והתקבל קובץ הנחיות הלחנה תקין, התוכנית תיצור רשימה עם המנגינה המולחנת ולאחר תוביל לתפריט שינוי הקבצים (התפריט שמתקבל באופציה 1 לאחר שמכניסים שם קובץ נכון).

### 3. תפריט הסיום

במידה והמשתמש הגיע לתפריט הסיום, התוכנית תבקש מהמשתמש שם קובץ לשמור בו את השמע, תשמור את השמע ותציג למשתמש את תפריט הכניסה.

### פרטים נוספים לגבי מימוש התוכנית

בכל שלב בתוכנית, אם המשתמש הכניס קלט לא תקין, נציג הודעת שגיאה אינפורמטיבית ולאחר נציג בשנית את ההודעה שהתוכנית הציגה לפני שהמשתמש הכניס קלט לא תקין.

### עבודה עם קבצי wav בתרגיל והקובץ wave\_helper.py

קבצי wav הם קבצי אודיו שאינם דחוסים. בכל קובץ יש שני מאפיינים המעניינים אותנו:

#### 1. קצב הדגימה (frame rate).

מספר דגימות השמע המייצגות שנייה.

בתרגיל: כאשר אנו משנים קובץ נתון, נשאיר את קצב הדגימה כפי שהיה במקור.

כאשר אנו יוצרים קובץ חדש כחלק מהלחנת מנגינה, קצב הדגימה יהיה 2000.

#### 2. רשימה המייצגת את השמע בקובץ.

האיברים ברשימה הם רשימות באורך 2 של מספרים שלמים שנעים בטווח שבין 32767 ל-32768.

האיברים ברשימה מסודרים לפי סדר השמעתם. כאמור, כל רשימה פנימית כוללת זוג מספרים שלמים (int), כאשר המספר הראשון בכל זוג מייצג את ערך הדגימה עבור הערוץ הראשון (אותו נשמע ברמקול הראשון/אוזנייה אחת), והמספר השני מייצג את ערך הדגימה עבור הערוץ השני (אותו נשמע ברמקול השני/אוזנייה השנייה). כאשר ניישם שינויים אותם ביקש המשתמש על רשימת השמע, נתייחס לכל אחד מהערוצים בפני עצמו, ללא תלות בערוץ השני.

לדוגמא, הרשימה הבאה  $[[1,2],[3,4],[5,6]]$  מייצגת 3 דגימות. כאשר ערוץ שמע 1 הוא הדגימות 1,3,5 והערוץ השני הוא 2,4,6.

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

לאורך התרגיל, כאשר נחשב ערכים שאינם שלמים אותם נרצה לכלול ברשימת השמע, נעגל ונמיר אותם בעזרת `int()`.

בשביל לפתור את התרגיל, תשתמשו בקובץ `wave_helper.py`.  
הפונקציות בקובץ תומכות בקריאת וכתיבת קבצי `wav` (ישנם מקרים נדירים בהם קבצי `wav` יהיו בפורמט בו הפונקציות אינן תומכות).  
הקובץ מממש את הפונקציות הבאות:

**`load_wave(wave_filename)`**

פונקציה זו מקבלת שם של קובץ `wave` (String). אם נתנו לפונקציה קובץ `wave` תקין, הפונקציה תחזיר 2 ערכים בסדר הבא: `[frame_rate, audio_data]`.  
כאשר `frame_rate` הוא `int`, `audio_data` היא רשימת השמע. כל ערך ברשימת השמע הוא רשימה של זוג `int`.  
במידה והייתה בעיה עם הקובץ שהוזן, הפונקציה תחזיר את המספר -1.

**`save_wave(frame_rate, audio_data, wave_filename)`**

פונקציה זו מקבלת את הפרמטר `frame_rate` `int`. הפרמטר `audio_data` צריך להיות רשימה של רשימות. הפונקציה שומרת את השמע שהועבר בקובץ `wave` ששמו (String) הוא `wave_filename`.  
למשתמש מותר לתת כל שם קובץ חוקי שירצה בו יישמר השמע. בפרט, שם הקובץ לא חייב לכלול את הסיומת `wav`.  
אם הקובץ נשמר בהצלחה, הפונקציה תחזיר 0.  
במידה והייתה בעיה עם הנתונים שהוזנו, הפונקציה תחזיר את המספר -1.

### שינוי קבצי `wav`

כאשר אנו מקבלים קובץ `wav` אותו המשתמש רוצה לשנות, ניתן להניח ששם הקובץ תקין (כלומר, לא צריך לבדוק ששם הקובץ מסתיים ב`wav`). בנוסף, ניתן להניח שאם התקבל קובץ `wav` תקין הוא נקרא בהצלחה ע"י `load_wav`.

ישנן 7 אופציות בתוכנית לשינוי קובץ `wav`.

כך נבצע את הפעולות על רשימת השמע אותה אנו מקבלים מ`wave_helper`:

1. היפוך השמע

נרצה להפוך את סדר הערכים ברשימה. לדוגמא, במידה והרשימה המייצגת את השמע שהתקבל היא:

`[[1,2], [2,3], [3,4], [4,5]]`

לאחר ההיפוך, הרשימה המייצגת את השמע תהיה:

`[[4,5], [3,4], [2,3], [1,2]]`

2. האצת מהירות השמע

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

בשביל להאיץ את מהירות השמע פי 2, ניקח מרשימה נתונה רק את הדגימות שהאינדקס שלהם זוגי. למשל, אם נקבל את רשימת השמע הבאה :

$[[1,1], [2,2], [3,3], [4,4], [5,5]]$

הרשימה אותה נחזיר תהיה :

$[[1,1], [3,3], [5,5]]$

3. האטת מהירות השמע

בשביל להאט את מהירות השמע פי 2, נוסיף בין כל זוג דגימות ברשימת השמע דגימה שהיא הממוצע של הדגימות הללו. לדוגמא, אם נקבל את הרשימה הבאה :

$[[10,10], [20,30], [30,50], [40, 60]]$

הרשימה שנחזיר אחרי ההאטה תהיה :

$[[10,10], [15, 20], [20,30], [25, 40], [30,50], [35, 55], [40, 60]]$

4. הגברת הווליום

נרצה להכפיל פי 1.2 את הערכים ברשימת השמע. נצטרך לדאוג שאנו לא עוברים את הערך המקסימלי והערך המינימלי של טווח המספרים החוקי (בין 32767 ל-32768-). במידה וקיבלנו ערך גבוה מהערך המקסימלי, נשים את הערך המקסימלי. במידה וקיבלנו ערך נמוך המערך המינימלי, נשים את הערך המינימלי. לדוגמא, כאשר הרשימה המייצגת את השמע שהתקבל היא :

$[[32767, 10002], [4, -2017], [0, 0], [-55, -55], [-100, -32760]]$

לאחר ההגברה, הרשימה המייצגת את השמע תהיה :

$[[32767, 12002], [4, -2420], [0, 0], [-66, -66], [-120, -32768]]$

5. הנמכת הווליום

נרצה לחלק ב-1.2 את הערכים ברשימה.

לדוגמא, כאשר הרשימה המייצגת את השמע שהתקבל היא :

$[[32767, 10002], [4, -2017], [0, 0], [-55, -55], [-100, -32760]]$

לאחר ההנמכה, הרשימה המייצגת את השמע תהיה :

$[[27305, 8335], [3, -1680], [0, 0], [-45, -45], [-83, -27300]]$

6. פילטר עמעום

בשביל לעמעם את השמע, נחליף את הערך של כל מספר במערך המייצג את השמע בממוצע של הערך הקודם, ערך המספר וערך המספר העוקב. כאשר העמעום של המספר הראשון במערך יהיה הממוצע של הערך הראשון והערך השני, והעמעום של הערך האחרון במערך יהיה הממוצע של הערך האחרון והערך הלפני אחרון. במידה והתקבל מספר שאינו שלם, נשתמש רק בחלק השלם של הממוצע שהתקבל.

לדוגמא, עבור הרשימה הבאה :

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

[[1, 1], [7, 7], [20, 20], [9, 9], [-12, -12]]

לאחר פעולת העמעום, הרשימה שתתקבל תהיה :

[[4, 4], [9, 9], [12, 12], [5, 5], [-1, -1]]

ניתן להניח כי רשימת שמע אותה נתבקש לעמעם כוללת לפחות שתי דגימות.

7. מעבר לתפריט הסיום

אם המשתמש בוחר לעבור לתפריט הסיום מבלי לבצע פעולות נוספות על השמע, השמע נשמר עם השינויים שביצענו בו עד כה. אם לא ביצענו שינויים בכלל, השמע ישמר כפי שהתקבל.

### הלחנת מנגינה

קצב הדגימה של קבצי wav אותם ניצור בחלק זה תמיד יהיה 2000. כזכור, קול הוא סוג של גל, וכל מספר במערך המייצג את הקול עבורנו הוא דגימה של הגל. לכן, כאשר אנו יוצרים שמע באורך שנייה, נצטרך ליצור 2000 דגימות לאורך גל כלשהו. החישוב לערך הדגימה ה- $i$  שלנו (עבור שני הערוצים) יהיה :

$$samples\_per\_cycle = \frac{sample\_rate}{frequency}$$
$$MAX\_VOLUME \times \sin\left(\pi \times 2 \times \frac{i}{samples\_per\_cycle}\right)$$

השתמשו במודול math בשביל לחשב את ערך פונקציית הסינוס והקבוע  $\pi$ . הווליום המקסימלי שלנו יהיה הערך המקסימלי שניתן לקבל – 32767. את ההמרה של התוצאה למספר שלם (בעזרת int()) נעשה אחרי שחישבנו את הביטוי.

התדר (frequency) עבור כל אחד מהתווים אותם אנו צריכים הוא :

שם התו באנגלית	תדר התו (frequency)	שם התו בעברית
A	440	לה
B	494	סי
C	523	דו
D	587	רה
E	659	מי
F	698	פה
G	784	סול

עבור שקט (Q), הערך של כל הדגימות הרלוונטיות יהיה תמיד 0.

הקלט עבור הלחנת מנגינה יתקבל כקובץ טקסט. הקובץ מפרט איזה תו נרצה לנגן (A,B,C,D,E,F,G,Q), וכמה זמן נרצה לנגן אותו. הזמן ייוצג כמספר שלם, המונה את מספר שש-עשריות (1/16) השנייה בהן נרצה לנגן את התו (שש-עשרית השנייה = 125 דגימות שמע, כאשר 2000 דגימות מייצגות שנייה). הפורמט יהיה תו, רווח, מספר שש-עשריות השנייה, רווח, תו, רווח מספר שש-עשריות השנייה... לדוגמא, אם נקבל את הקלט הבא :

# בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

F 16 G 8 Q 32 G 1

ניצור רשימת שמע המנגנת את התו F במשך שנייה, את התו G במשך חצי שנייה, שתי שניות של שקט ובסוף ננגן את התו G במשך שש-עשר שניות השנייה.

תחילת רשימת השמע תיראה כך :

[[0, 0], [26629, 26629], [-31033, -31033], [9536, 9536], ...]

בנוסחה לחישוב ערך הדגימה, אנו מסתמכים על האינדקס (i) של הדגימה ברשימת השמע. עבור כל מופע חדש של תו בקובץ הקלט, נאפס את i. כלומר, עבור הקלט 1 F 1 F, בדגימה שניצור במקום ה-126 (רשימת השמע באינדקס 125), i יהיה שווה ל-0 והדגימה תהיה [0, 0]. לעומת זאת, עבור הקלט 2 F, בדגימה שניצור במקום ה-126, i יהיה שווה ל-125 והדגימה תהיה [-23169, 23169].

קובץ קלט תקין יכול לכלול רצף המורכב אך ורק מהתווים A,B,C,D,E,F,G,Q ומשך הזמן המוצג במספרים שלמים. קובץ קלט תקין יכול לכלול ירידות שורה לפני, באמצע או בסוף ההנחיות. קובץ קלט תקין יכול לפחות תו אחד. ניתן להניח שהתוכנה מקבלת קובץ קלט תקין.

ניתן למצוא דוגמאות נוספות לקבצי קלט תקינים וקבצי הפלט המתקבלים ע"י התוכנה בתיקייה composition samples במודל.

## בדיקת התוכנית

אנו ממליצים לכתוב את התוכנית בשלבים. לאחר כל פונקציה שתכתבו, הקפידו לבדוק אותה ע"י קריאה עם פרמטרים שונים והשוואת התוצאות שקיבלתם למה שמצופה. חשבו על קלטים שונים לפונקציות שיכולים לגרום לתוצאות שונות, כולל מקרי קצה. אנו מאוד ממליצים שלא לכתוב את כל התוכנית יחד ולבדוק אותה בשלמותה בסוף, מכיוון שהדבר יקשה עליכם מאוד במציאה וטיפול בבעיות. כחלק ממעבדה 6 תלמדו כלים שיעזרו לכם לכתוב טסטים עבור התוכנה.

## קבצים לדוגמה

ניתן למצוא קבצי wav עליהם ניתן לעבוד בתיקייה "wav\_samples" במודל. דוגמאות לקבצי הלחנה וקבצי wav התואמים ניתן למצוא בתיקייה "composition\_samples" במודל.

## הוראות הגשה

את תרגיל זה תגישו בזוגות.

לפני ההגשה ותחת הלינק המיועד להגשה, עליכם לפתוח קבוצה ב-moodle. אחד השותפים ייצור את הקבוצה על ידי הזנת שם השותף השני (שימו לב להוסיף את השם בדיוק כפי שהוא מופיע ב-moodle כולל אותיות גדולות וקטנות במקומות הנכונים). השותף (שלא יצר את הקבוצה) יוכל לראות שרשמו אותו

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

על ידי כניסה לקישור ההגשה וצפייה בשם בן הזוג. כדאי לרשום את בן הזוג בשלב מוקדם (אין צורך להגיש את התרגיל בפועל על מנת להירשם כזוג). כמו כן, וודאו כי הקבוצה אינה מכילה יותר משני שותפים. עליכם להגיש קובץ נוסף בשם AUTHORS (ללא כל סיומת). קובץ זה יכול שורה אחת ובו הלוגינים של שני הסטודנטים המגישים, מופרד ע"י פסיק. כך :

minniemouse,mickeymouse

(תקף רק למי שחוזר על הקורס): במידה והגשתם את התרגיל לבד, עליכם להוסיף את קובץ ה-AUTHORS עם הלוגין שלכם בלבד (ללא פסיק).

ודאו כי הגשתם קובץ **AUTHORS** תקין! אי הגשה של קובץ **AUTHORS** תקין תגרור הורדה בציון. עליכם להגיש את הקובץ ex6.zip (בלבד) בקישור ההגשה של תרגיל 6 דרך אתר הקורס על ידי לחיצה על "Upload file". אנו ממליצים להתחיל לעבוד על התרגיל בשלב מוקדם שכן התרגיל ארוך מקודמיו. ex6.zip צריך לכלול את הקבצים:

1. wave\_editor.py
2. description.txt - כולל הסברים על החלוקה הפנימית של הקוד שלכם לפונקציות ולקבצים נוספים (אם יש).
3. AUTHORS – הכולל את הלוגינים של מגישי התרגיל בפורמט שתואר.

### הנחיות כלליות בנוגע להגשה

- הנכם רשאים להגיש תרגילים דרך מערכת ההגשות באתר הקורס מספר רב של פעמים. ההגשה האחרונה בלבד היא זו שקובעת ושתיבדק.
- לאחר הגשת התרגיל, ניתן ומומלץ להוריד את התרגיל המוגש ולוודא כי הקבצים המוגשים הם אלו שהתכוונתם להגיש וכי הקוד עובד על פי ציפיותיכם.
- באחריותכם לוודא כי – PDF הבדיקות נראה כמו שצריך.
- קראו היטב את קובץ נהלי הקורס לגבי הנחיות נוספות להגשת התרגילים.
- שימו לב - יש להגיש את התרגילים בזמן!

### ביקורת עמיתים (Peer review)

כחלק מתרגיל 8, תתבקשו לכתוב משוב על פתרון של תרגיל 6 שנכתב ע"י אחד מהסטודנטים בקורס. בצורה דומה, אחד מהסטודנטים בקורס ייתן משוב על התרגיל אותו כתבתם. ביקורת העמיתים תערך אחרי שתקבלו ציון ולא תשפיע על ציונכם בתרגיל 6. עם זאת, כאשר אתם כותבים את התרגיל קחו בחשבון שסטודנטים אחרים בקורס יקראו וינסו להבין את הקוד שלכם, כך שהקוד צריך להיות מובנה וברור. **שימו לב שאתם לא כותבים את מספר תעודת הזהות שלכם באף קובץ אותו אתם מגישים.** הקובץ יהיה זמין לצפייה לסטודנט שיכתוב משוב על התרגיל שלכם.

בהצלחה!