

## מבוא

ברכות! הגעתם לסוף הקורס "מבוא למדעי המחשב", וכעת נותר לכם להדגים את הידע שצברתם במהלך הקורס. הפרוייקט הוא המטלה המסכמת של הקורס ובו תדגימו את כל המיומנויות שרכשתם במהלך הסמסטר.

## הנחיות כלליות

- הפרוייקט הוא **אישי**, כלומר על כל סטודנט לעבוד על הפרוייקט בעצמו. אי אפשר לעבוד בזוגות או בקבוצות.
- היות והפרוייקט הוא המטלה המסכמת של הקורס אנו מדגישים שהעבודה על הפרוייקט חייבת להיות **עצמאית לגמרי**. **אין לדון עם סטודנטים אחרים על הפרוייקט שלכם** (בשונה מהתרגילים בהם מותר לדון לגבי הקווים הכלליים), אין לחלוק או להעביר קוד בין סטודנטים, ועליכם לדבג ולכתוב את הפרוייקט לגמרי לבד. באופן דומה לא תינתן תמיכה של ה-lab support או של סגל הקורס לגבי הפרוייקט.
- בשונה מיתר התרגילים בקורס, אנחנו לא מגדירים לכם את ההתנהגות המדוייקת של התוכנית או אילו מחלקות/פונקציות לממש. עליכם להחליט על כך בצורה עצמאית.
- בהמשך הקובץ מוצגות 5 אפשרויות לפרוייקט, שמהן עליכם לבחור באפשרות אחת בלבד שאותה תגישו. אין להגיש יותר מפרוייקט אחד (אם יוגשו יותר תיבדק רק הגשה אחת שתיבחר באופן שרירותי).
- שימו לב! בדומה לשאר התרגילים בקורס, אינכם רשאים להעתיק קוד מסטודנטים אחרים, מהאינטרנט, או ממקורות אחרים. אבל, בניגוד לשאר התרגילים בקורס, במהלך העבודה על הפרוייקט אתם רשאים להשתמש בכלי בינה מלאכותית (כגון ChatGPT, CoPilot וכו') כרצונכם. עם זאת, חשוב להדגיש שפעמים רבות כלים אלו כותבים תוכניות מחשב עם שגיאות ועל כן מומלץ בחום לא להסתמך עליהם באופן מוחלט.
- בנוסף להגשת התוכנית שכתבתם יהיה עליכם גם להגיש סרטון הדגמה קצר על מבנה התוכנית, רכיבים מרכזיים והרחבות שבחרתם לממש.

## הנחיות כלליות לגבי כתיבת הקוד

- הקפידו לעבוד בהתאם לעקרונות שנלמדו בכיתה. בפרט, הקפידו על יעילות, על מבנה נכון של הקוד ועל סגנון הכתיבה (תיעוד, שמות משתנים משמעותיים, שימוש בקבועים וכו').
- עליכם להשתמש בתכנות מונחה עצמים לצורך כתיבת הפרוייקט.
- הקוד שלכם צריך לטפל היטב בשגיאות. התוכנית לא אמורה להתרסק לעולם גם אם קיבלה קלט לא כמצופה, כשלה בפתיחת קובץ, או לא הצליחה להתחבר לשירות כלשהוא.
- בעת כתיבת הפרוייקט יש להשתמש ב-python3.9, ועליכם לוודא שהקוד שלכם ניתן להרצה על מחשבי בית הספר.

## הדרישות בכל פרויקט, הרחבות אפשריות, ותרומתן לציון

- בתיאור כל פרויקט מוצגות דרישות מינימום שיש לממש (בהתאם להנחיות ולדגשים הכלליים בחלק הראשון של המסמך) על מנת לקבל ציון עובר בפרויקט.
- בנוסף מתוארות הרחבות נוספות שניתן לממש על מנת להעלות את הציון.
- ההרחבות הן המלצות בלבד, ונועדו לתת לכם רעיונות. מומלץ לחשוב על הרחבות מעניינות ושימושיות משלכם שעשויות גם הן להעלות את הציון.
- אין ציפיה שתממשו את כל ההרחבות המוצעות לפרויקט ואין צורך לעשות זאת (חלקן עשויות להיות קשות יותר מאחרות).
- בנוסף להרחבות הספציפיות עבור כל פרויקט, גם הפריטים הבאים יחשבו להרחבות שניתן לקבל עליהן ניקוד:

– כתיבת טסטים ב `pytest`.

אנחנו נריץ את הפקודה `pytest` בתיקיית הפרויקט כדי להפעיל את הטסטים. ייתכן שגשתם גם ב- [pytest-cov](#) על מנת להעריך את רמת הכיסוי של הטסטים שכתבתם (כמה מהקוד שכתבתם נבדק בפועל על ידי הטסטים).

– שימוש ב- `type annotations` (שייבדק באמצעות הרצת `mypy` על קבצי הפרויקט).

- הציון הסופי בפרויקט יקבע ע"י כלל ההרחבות שמימשתם בפרויקט, לפי דרגת הקושי של כל הרחבה ולפי טיב המימוש שלכם (מודולריות, כתיבת קוד נקי, תיעוד, וכד').
- כדי לשמור על הוגנות בין הפרויקטים השונים והרחבות שונות ברמת קושי שונה, נוודא שציוני הפרויקטים יהיו דומים לציוני המבחנים בקורס בשנים קודמות.

## שימוש בספריות חיצוניות

מותר להשתמש בכל ספריית פייתון חיצונית שניתן להתקין בעזרת `pip` (או שזמינה ללא התקנה בפייתון). אסור להשתמש בספריות שפותרות ישירות את הבעיה שב"ליבת הפרויקט" (ספריות אסורות כאלו ניתנות לדוגמא בתיאורי הפרויקטים).

כדי להתקין ספריות העזרו בתהליך שלמדנו בקורס:

1. פתחו סביבה וירטואלית, באמצעות הפקודה (החליפו את `<env_name>` בשם כלשהו לבחירתכם):

```
python3 -m venv <env_name>
```

2. הפעילו את הסביבה הוירטואלית, באמצעות הפקודה:

```
source <env_name>/bin/activate
```

אם הפקודה לא עובדת נסו את הפקודה:

```
source <env_name>/bin/activate.csh
```

3. נסו להתקין את הספרייה באמצעות הפקודה:

```
pip install <package>
```

כמו שמפורט בהמשך (תחת נהלי ההגשה), עליכם להגיש קובץ `requirements.txt` שמכיל את הספריות הדרושות לצורך הרצת הפרויקט. לצורך יצירת הקובץ, ניתן ליצור סביבה וירטואלית כמפורט לעיל, להפעיל אותה ולהתקין עליה את

הספריות הנדרשות. לאחר שוודא אתם שכלל הספריות הנדרשות מותקנות (כדאי להריץ את הפרויקט ולראות שהוא רץ ללא שגיאות), ניתן להריץ את הפקודה:

```
pip freeze > requirements.txt
```

פקודה זו תיצור לכם את הקובץ requirements.txt בתיקייה שבה אתם נמצאים, בפורמט הנדרש.

ניתן גם להתקין את הספריות בקובץ requirements.txt ע"י הפקודה:

```
pip install -r requirements.txt
```

מומלץ לבצע התקנה בסביבה וירטואלית חדשה לפני ההגשה כדי לוודא שקובץ requirements.txt אכן כולל את כל הספריות הדרושות.

## יצירת סרטון הסבר על הפרויקט

כחלק מהפרויקט, אתם מתבקשים להגיש סרטון שבו אתם מציגים את הפרויקט ומסבירים על האופן בו מימשתם אותו. הסרטון מהווה חלק מהציון. מטרת הסרטון היא לסייע לנו להעריך נכון את ההשקעה שלכם ואת טיב הפרויקט ולמקד את הבדיקה במה שאתם חושבים שהיווה את החלקים המשמעותיים והמאתגרים בבנייתו. על כל סטודנט להקליט את הסרטון בקולו, והוא יכול להיות בעברית או באנגלית. הקפידו לדבר בצורה ברורה ולהקליט בסביבה שקטה.

על הסרטון להיות באורך של עד 7 דקות (זו מגבלה קשיחה), והוא צריך לכלול את המרכיבים הבאים:

- שקופית פתיחה- שם הפרויקט אותו בחרתם, שם הסטודנט ושם המשתמש שלכם (CSE user), ת"ז של המגיש (או מספר סטודנט).
- דוגמאות הרצה של התוכנית הכוללות הקלטה של המסך. יש להדגים את כל ההרחבות המרכזיות שמימשתם וכן להראות רעיונות מקוריים או אלגוריתמים מעניינים שבהם עשיתם שימוש.
- תיאור קצר של מבנה התוכנית וסיור בחלקים המרכזיים של הקוד (אין צורך להראות את הכל).
- השתדלו להראות לנו שהשתמשתם בעקרונות שנלמדו בקורס בצורה נכונה, ושכתבתם קוד יעיל, נקי ובנוי היטב.
- הקפידו שהסרטון יהיה ענייני, מקיף ויסודי. הפעילו שיקול דעת ואל תתעכבו על דברים שוליים.

לצורך יצירת הסרטון כדאי ליצור מצגת שתכיל את כלל הרכיבים הדרושים ובמהלך הסרטון להציג אותה.

על מנת להקליט את הסרטון, תוכלו לפתוח שיחת zoom (ללא משתתפים נוספים מלבדכם) שבה תציגו את המצגת, הקוד, וריצת התוכנית.

לחילופין, אם השתמשתם ב-PowerPoint לצורך יצירת המצגת, אנו ממליצים להקליט את הצגת הפרויקט בתוך powerpoint ולבסוף לייצא את המצגת לוידאו (File > Export > Create a Video).

כמו כן, ניתן להשתמש ב-PowerPoint גם לצורך הקלטת המסך (Insert > Screen Recording).

## הגשת הסרטון

יש להעלות את הסרטון ל-YouTube בחשבון הגוגל האישי או האוניברסיטאי שלכם. לסרטוני YouTube יש שלוש אפשרויות להגדרת גישה: public, private, unlisted. קבעו את הגדרת הגישה כ-unlisted כך שכל מי שיש לו קישור לסרטון יוכל לגשת אליו אך הסרטון לא יחשף למי שאין לו קישור ישיר. עליכם לוודא שהקישור נפתח עבור משתמשים אחרים (פתחו דפדפן לגלישה "פרטית" או "incognito" וודאו שהקישור נפתח). את הקישור יש לצרף בתוך קובץ הנקרא VIDEO (ללא סיומת) שיצורף לקבצי ההגשה.

## נהלי הגשה

- עליכם להגיש קובץ zip בשם **final\_project.zip** והוא צריך להכיל את הקבצים הבאים:
  - requirements.txt - קובץ שמכיל את כלל החבילות שנדרשות לצורך הרצת הקוד, כמו שמוסבר לעיל.
  - README (קובץ טקסט אבל ללא סיומת txt) - שבו יופיעו:
    - רשימה של ההרחבות שעשיתם (משפט אחד לכל הרחבה). רשמו קודם את ההרחבות הגדולות והמורכבות יותר כדי שנדע להתמקד בהן בזמן הבדיקה. תארו בקצרה איך להפעיל כל הרחבה.
    - (אופציונלי) הסבר על 2-3 הרחבות מעניינות (פסקה לכל אחת).
  - קובץ בשם VIDEO (גם ללא סיומת) שמכיל שורת טקסט בודדת ובה הקישור המלא לסרטון ה-YouTube שמציג את הפרויקט שלכם (שימו לב שהסרטון שלכם מוגדר נכון בהרשאת גישה unlisted).
  - הקישור המלא נראה בערך כך: <https://www.youtube.com/watch?v=y8KyI0WNg40>
  - כלל קבצי הקוד הדרושים להרצת הפרויקט, בהתאם לנהלים הספציפיים עבור הפרויקט אותו בחרתם. בין קבצי הקוד עליכם לכלול גם קובץ בשם **main.py** שהוא הקובץ הראשי שדרכו מריצים את התוכנית.
  - ההגשה מתבצעת דרך אתר הקורס, בקישור הייעודי עבור הפרויקט אותו בחרתם. שימו לב שסקריפט קדם ההגשה (presubmission) יכלול בדיקות בסיסיות בלבד הנוגעות אך ורק לתקינות הקבצים שהוגשו.
  - כמו בתרגילים שהגשתם לאורך הקורס - גם כאן ניתן להגיש מספר רב של פעמים וההגשה האחרונה בלבד היא זו שקובעת ותיבדק.
  - וודאו היטב שאתם מגישים את הקבצים הנכונים ושניתן להתקין את הספריות ולהריץ את הקוד שלכם בקלות.
- השלבים הדרושים להרצת הקוד (מומלץ מאוד לבדוק זאת על מחשבי בית הספר) הם:
  - יצירת תיקייה שמכילה את final\_project.zip וחילוף הקבצים מקובץ ה-`zip`.
  - יצירת סביבה וירטואלית והפעלתה (ראו הנחיות לעיל).
  - התקנת הספריות מהקובץ requirements.txt בסביבה הוירטואלית, באמצעות הפקודה:  
`pip install -r requirements.txt`

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

- הרצת הקוד: הקובץ הראשי של כלל הפרוייקטים יקרא `main.py`

לאחר הפעלת הסביבה הוירטואלית, הפקודה הבאה תרוץ לכל פרוייקט (לא משנה מאיזה סוג) ותיתן הסבר על איך להפעילו.

```
python main.py --help
```

### שימו לב!

פרוייקטים שלא יהיו ניתנים להרצה (בעקבות בעיה בקוד/ בעיה עם התקנת הספריות/ שמות שגויים של קבצים וכד'), או פרוייקטים שלא מכילים את הקבצים הדרושים (קובץ README וסרטון) יקבלו ציון נמוך ללא אפשרות לערער.

**הגשות מאוחרות:** הגשה מאוחרת תגרור הורדה של 7 נק' בציון לכל יום איחור (ההורדה תהיה רציפה כך שאיחור קטן יותר יגרור הורדה פרופורציונלית לכך).

### אמינות ויושרה אקדמית

אנו חוזרים ומדגישים כי העבודה על הפרוייקטים היא אישית. אין לשתף קוד בכל צורה שהיא (ובאחריותכם לוודא שלסטודנטים אחרים אין גישה כלשהי לקוד שלכם), אין להביט בפתרון של סטודנטים אחרים ואין לעשות שימוש בפתרונות דומים מרשת האינטרנט.

הפרויקט מהווה חלק משמעותי מהציון הסופי בקורס, ועל כן נקפיד הקפדה יתרה על אכיפה של העתקות.

**אנא אל תפגעו באמון שאנחנו נותנים בכם.**

### בהצלחה!

בהמשך המסמך מוצגות 5 האפשרויות לפרוייקטים, מתוכן עליכם לבחור באפשרות אחת.

## אפשרות 1 - תוכנה לכיווץ קבצי טקסט

### תיאור הפרויקט

בפרויקט זה תממשו תוכנה שמכווצת קבצים רבים לקובץ יחיד (בדומה לתוכנה ליצירת קבצי zip עליה תוכלו לקרוא [כאן](#)). לצורך כיווץ הקבצים, התוכנה תשתמש באלגוריתם [דחיסת נתונים](#) ידוע.

### דרישות לקבלת ציון עובר

- מימוש אלגוריתם כיווץ ופתיחת קובץ על פי [run length encoding](#) עם תמיכה באפשרות לקבוע את גודל יחידת הכיווץ (כלומר מה אורך היחידה החוזרת על עצמה בכיווץ), החל מ byte בודד ועד מספר כלשהוא n של bytes.
- יש לכתוב את קובץ הארכיב לדיסק באופן בינארי - "wb". אין לעשות שימוש ב-pickle או בפתרון דומה שמבצע סריאליזציה של אובייקטים בפייתון.
- תמיכה בכיווץ מספר קבצים לקובץ ארכיון בודד (מספר הקבצים הניתנים כקלט יכול להשתנות), וחילוץ קבצים מקובץ הארכיון.
- תמיכה בכיווץ קבצים בינאריים (לא רק קבצי טקסט). שימו לב שזה יצריך עבודה עם מחרוזות בינאריות במקום מחרוזות טקסט רגילות. ראו [כאן](#), [וכאן](#).
- תמיכה בכיווץ ובפתיחה של תיקייה המכילה מספר קבצים (ללא תיקיות פנימיות). בעת הפתיחה תיווצר התיקייה המכילה את הקבצים עם כל תוכנה.
- הוספה של קובץ חדש לקובץ ארכיון קיים.
- אפשרות לראות מידע נוסף על טיב הכיווץ, כלומר בכמה התכווץ כל קובץ וכמה זמן לקחה פעולת הכיווץ או הפתיחה של הארכיון.
- אפשרות לבדוק שהקובץ "תקין" ואין שגיאה בקידוד שלו. למשל קובץ שלא עומד בפורמט שהגדרתם לקובץ מכווץ צריך להחזיר שגיאה מתאימה. חישבו היטב על מה קורה כשפותחים קובץ שאינו תקין. האם מוחקים או מדלגים על חלק מהקבצים? מה הודעות השגיאה?

### רעיונות להרחבות אפשריות

- כיווץ/פתיחה של תיקייה המכילה מספר קבצי טקסט/תיקיות באופן רקורסיבי (בדומה ל zip -r)
- הוספת תמיכה בצפייה בתוכן של ארכיון (בדומה ל unzip -l)
- הצפנת הקובץ המכווץ והוספת סיסמא שתידרש על מנת לפתוח אותו (בדומה ל zip -e)
- עדכון קבצים קיימים בארכיון ו/או מחיקת קבצים מתוך ארכיון קיים (בדומה ל zip -u או zip -d)
- הוספת אפשרות להתעלם מקבצים מסוימים/סוג קובץ מסוים/תיקיות מסוימות בזמן הכיווץ (בדומה ל zip -x)
- מימוש אלגוריתמי כיווץ ופתיחת קובץ נוספים למשל - [קוד האפמן](#) ו/או [למפל זין](#)
- בניית ממשק משתמש גרפי (GUI) לתוכנית הכיווץ שתאפשר לבחור בקלות קבצים לכיווץ.

### דגשים נוספים

- לצורך הטיפול בארגומנטים הניתנים משורת הפקודה, מומלץ להשתמש בספריה [argparse](#).
- אין להשתמש בספריות כיווץ קיימות בפייתון כגון [zipfile](#) או אחרות (היוצרות קבצי rar, tar, gz).



## אפשרות 2 - תוכנה ליצירה ועריכת גרפיקה וקטורית

### תיאור הפרויקט

בפרויקט זה תממשו תוכנה המאפשרת למשתמש להכין ולערוך תרשימים ויצירות דיגיטליות ביצוג וקטורי באמצעות ממשק גרפי ידידותי ונוח (כמו inkscape או Adobe Illustrator).

### דרישות לקבלת ציון עובר

- יצירת ממשק גרפי (GUI) שבאמצעותו המשתמש יוכל לבצע את כלל הפעולות המופיעות להלן. הממשק הגרפי צריך להיות כמה שיותר אינטואיטיבי ונוח (כלומר שמשתמש ממוצע יוכל להבין איך לעבוד עם התוכנה ללא הסברים נוספים).
- יצירת לוח שעליו המשתמש יכול לצייר.
- אובייקטים שצוירו על הלוח ניתנים להזזה והסרה.
- ניתן לשנות תכונות ויזואליות של כל אובייקט (צבע, עובי קו, מילוי).
- סדר האובייקטים על הלוח ניתן לשינוי (איזה אובייקט מצויר מעל איזה).
- הוספת כלים בסיסים לציור: ציור צורות יסוד (עיגול, מרובע, משולש), ציור חופשי של קו, ציור פוליגון בעזרת נקודות, מחק.
- בחירת עובי הקו וגודל המחק.
- אפשרות להוספת טקסט בגדלים שונים, גופנים שונים וצבעים שונים.
- אפשרות לשמור את היצירה שהמשתמש הכין ולהמשיך בעריכה.
- אפשרות לייצא את התרשים הווקטורי לפורמט גרפי פשוט (jpeg, gif או משהו דומה).
- אפשרות למחוק את הלוח ולהתחיל ציור חדש.

### רעיונות להרחבות אפשריות

- הוספת אפשרות לבחור מספר אובייקטים ולהזיז אותם יחד.
- קיבוץ מספר עצמים לקבוצה, פירוק הקבוצות.
- מילוי בצבע של אובייקט סגור / איזור תחום על ידי קו.
- יצירת אפקטים למילוי צורות (מילוי הדרגתי / מילוי עם מרקם מסוים).
- אפשרות לבטל את הפעולות האחרונות שבוצעו (undo), ולבצע אחזור פעולה (redo).
- תמיכה בפורמטים וקטוריים סטנדרטיים (pdf, svg, eps).
- תמיכה ברוטציה ושיקוף של אלמנטים.
- תמיכה בהעתקה והדבקה.

### דגשים נוספים

- על מנת ליצור את הממשק הגרפי מומלץ להשתמש בספריה [Tkinter](#).
- כמו כן, ניתן להשתמש בספריה Pillow לצורך פתיחה ושמירה של תמונות.



## אפשרות 3- משחק דמקה סינית

### תיאור הפרויקט

בפרויקט זה תממשו משחק [דמקה סינית](#). על התוכנית להציג את התנהלות המשחק, על כלל חלקיו וחוקיו השונים.

### דרישות לקבלת ציון עובר

- מימוש המשחק הבסיסי של דמקה סינית (ראו [כאן](#)) עם ממשק טקסטואלי.
- דחייה של מהלכים לא חוקיים של המשתמשים.
- תמיכה במשחק בין 2/3/4/6 משתמשים לוקאליים (על אותו המחשב). לכל משתמש צריכים להיות מספר פרטים: שם, מספר ניצחונות והפסדים, וצבע או סימן של הכלים שלו על הלוח.
- תמיכה במשחק מול שחקנים ממוחשבים "פשוטים" אשר מבצעים מהלך שרירותי חוקי בכל תור.
- תמיכה באפשרות יצירת קבצי Log במהלך המשחק.
- קובץ log הוא קובץ טקסט שמציין את התנהלות המשחק בכל תור (השעה שבה התור התרחש, מי השחקן, מה המהלך שבוצע וכד').
- אפשרות לטעון קבצי Log של משחקים קודמים ולצפות במצב הלוח אחרי כל מהלך. במידה והמשחק לא הסתיים- יש לאפשר להמשיך אותו מהנקודה שבה הופסק.

### רעיונות להרחבות אפשריות

- מימוש GUI – ממשק משתמש גרפי ללוח המשחק דרכו ינתנו מהלכי השחקנים ויוצג הלוח.
- מימוש וריאנטים של המשחק כגון הווריאנט Capture (ראו [פה](#)).
- מימוש אסטרטגיה של שחקן מחשב מתוחכם יותר- ניתן לממש 2-3 אסטרטגיות שונות.
- תמיכה באפשרות של טורניר עם קבוצה גדולה של שחקנים.
- תמיכה במשחק עם לוח בגודל דינאמי.
- הוספת אפקטים של אודיו וסאונד למהלכים שונים של המשחק.
- אפשרות לשחק מול שחקן אחר ברשת האינטרנט.

### דגשים נוספים

- ניתן להשתמש בספרייה [Tkinter](#) עבור יצירת הממשק הגרפי.

## אפשרות 4- תוכנה ליצירת גיליונות אלקטרוניים

### תיאור הפרויקט

בפרויקט זה תממשו תוכנה המאפשרת למשתמש ליצור ולנהל גיליונות אלקטרוניים, להזין לתוכן נתונים, לבצע חישובים ועוד, בדומה לתוכנה Excel או Google Sheets.

### דרישות לקבלת ציון עובר

- מימוש התוכנה כך שהמשתמש יוכל ליצור גיליון, להזין לתוכן נתונים (מספריים או טקסטואליים) ולבצע חישובים תוך שימוש בנתונים שקיימים בגיליון.
- תמיכה בנוסחאות עם פעולות חשבון בסיסיות (חיבור, חיסור, כפל וחילוק) בין תאים שונים. כלומר, אפשרות להשתמש בשמות התאים כשמבצעים חישוב כלשהו (לדוגמא, החישוב  $A1*3$  צריך להחזיר את המכפלה של 3 עם הערך המופיע בתא A1).
- תאי נוסחה צריכים להתעדכן אוטומטית כשהתאים בהם הם תלויים התעדכנו.
- יישום פונקציות בסיסיות: MIN, MAX, SUM, AVERAGE הפועלות על טווח כלשהו של תאים.
- טיפול מתאים בשגיאות - הצגת הודעות שגיאה אינפורמטיביות במקרה של שגיאות בנוסחאות או קלטים לא תקינים.
- תמיכה בשמירה וטעינה של גיליונות בפורמט שהוא human readable כגון json או yaml.

### רעיונות להרחבות אפשריות

- מימוש GUI – ממשק משתמש בו יוצג וייערך הגיליון האלקטרוני.
- תמיכה בגיליונות מרובים בלשוניות (tabs) ואפשרות לעבור בין לשוניות ולגשת מגיליון אחד לערכים ששמורים בלשונית אחרת.
- עיצוב תאים - לאפשר למשתמש להתאים אישית את עיצוב התאים (צבע, גופן וכד').
- יישום פונקציות מתקדמות - מתמטיות וסטטיסטיות (לדוגמא, SQRT, COUNTIF, IF וכד').
- הוספת אפשרות לבטל פעולות אחרונות שבוצעו או לבצע מחדש פעולות שבוטלו (undo, redo).
- תמיכה בייבוא נתונים ממקורות חיצוניים (כגון קבצי CSV).
- תמיכה בייצוא גיליונות לפורמטים שונים (CSV, אקסל, PDF, ועוד).
- מילוי אוטומטי של תאים באמצעות גרירה של תא סמוך המכיל נוסחה כלשהי.

### דגשים נוספים

- ניתן לממש את התוכנה באמצעות ממשק גרפי למשתמש (GUI) או באמצעות הצגה על הטרמינל. על מנת ליצור ממשק גרפי, ניתן להשתמש בספרייה [Tkinter](#).
- על מנת להציג את התוכנה בטרמינל, ניתן להשתמש גם בספרייה [curses](#) (תוכלו להיעזר גם בסדרת הסרטונים [הזו](#) כדי ללמוד עליה עוד).
- על מנת לנהל ולשמור את הנתונים, ניתן להשתמש בספרייה [pandas](#).

## אפשרות 5 - סימולציה של הילוך מקרי

### תיאור הפרויקט

בפרויקט זה תממשו תוכנת סימולציה עבור הילוכים מקריים שונים, וכלים לניתוח וויזואליזציה שלהם.

### דרישות לקבלת ציון עובר

- תכנון פורמט לקלטים מתאימים (פרמטרים עבור הסימולציה) שנלקחים משורת הפקודה ו/או מקובץ קונפיגורציה. (ראו למשל את [argparse](#)).
- הסימולציה תכלול אפשרות לבחירה בין מהלכים (walkers) שונים ולשינוי המרחב בו הם מהלכים.
- סוגי המהלכים (walkers) בהם יש לתמוך:
  1. בכל צעד נבחר כיוון אקראי (בזווית כלשהיא) ויבוצע צעד של יחידה אחת בכיוון זה.
  2. צעד בכיוון אקראי, אבל אורך הצעד נדגם באקראי מהתפלגות אחידה בין  $[0.5, 1.5]$ .
  3. דומה למהלך 1, אבל בחירת הכיוון נעשית רק בין 4 אפשרויות דיסקרטיות: למעלה, למטה, ימינה, או שמאלה.
  4. מהלך מוטה: בוחר כיוון בהסתברות לא אחידה – יש סיכוי מוגבר להילוך שמוטה לכיוון מסוים (למעלה, למטה, ימינה, שמאלה, או בכיוון ראשית הצירים).
- הסימולציה תערך במישור דו-מימדי אינסופי. אפשר יהיה להוסיף לסימולציה את הדברים הבאים:
  - מכשולים (שלמהלך המקרי אסור לעבור אותם).
  - שערים מכושפים שאם המהלך המקרי עובר בהם / נכנס לתחומם הוא מועבר לנקודה אחרת במישור (מיקום השערים והיעד שלהם יקבעו כפרמטרים של הסימולציה).
- ניתן לחזור על כל סימולציה מספר רב של פעמים ולשמור סטטיסטיקות על הגדלים הבאים (בממוצע על פני הרבה סימולציות) בקבצים:
  - המרחק הממוצע של המהלך מהראשית לאחר  $N$  צעדים.
  - הזמן הממוצע שלקח למהלך לצאת מרדיוס 10 יחידות מסביב לראשית.
  - המרחק הממוצע מציר  $x$  או  $y$  לאחר  $N$  צעדים.
  - מספר הפעמים שהמהלך חצה את ציר  $y$  לאחר  $N$  צעדים (בממוצע על פני הריצות).
- יצירת גרפים המציגים את הסטטיסטיקות הנ"ל כפונקציה של מספר הצעדים ושמירת הגרפים בקובץ לפי בקשת המשתמש. (ראו למשל את ספריות `matplotlib`, `seaborn` ואחרות).

### רעיונות להרחבות אפשריות

- הוספת סוגי מהלכים נוספים, שחרור כמה מהלכים בו זמנית עם "משיכה" / "דחיה" ביניהם.
- אפשרות ל `restart` - בכל נקודת זמן יש למהלך מקרי הסתברות מסוימת לחזור לראשית (ההסתברות הזו יכולה להיות קבועה או תלויה בתנאים מסוימים).
- אלמנטים אחרים במרחב ההילוך המקרי כמו למשל אזורים בהם המהלכים איטיים יותר.
- הרחבה למרחב תלת מימדי.

- מימוש יצירת קריסטלים בעזרת מהלכים מקריים (diffusion limited aggregation, ראו [פה](#)), וויזואליזציה [פה](#)). כולל סוגי מהלכים שונים עם אינטראקציות "הידבקות" שונות ביניהם, הרחבה לתלת מימד (ראו וויזואליזציה [פה](#)).
- בניית ממשק משתמש גרפי (GUI) לתוכנית הסימולציות שתאפשר בחירת פרמטרים שונים ואופציות שונות עבור הסימולציה, וויזואליזציה מובנית של ההילוכים ושל הגרפים הנלווים. אפשרות: שליטה דינמית ושינוי פרמטרים תוך כדי הסימולציה.