תרגיל בית לקורס ממ"ס

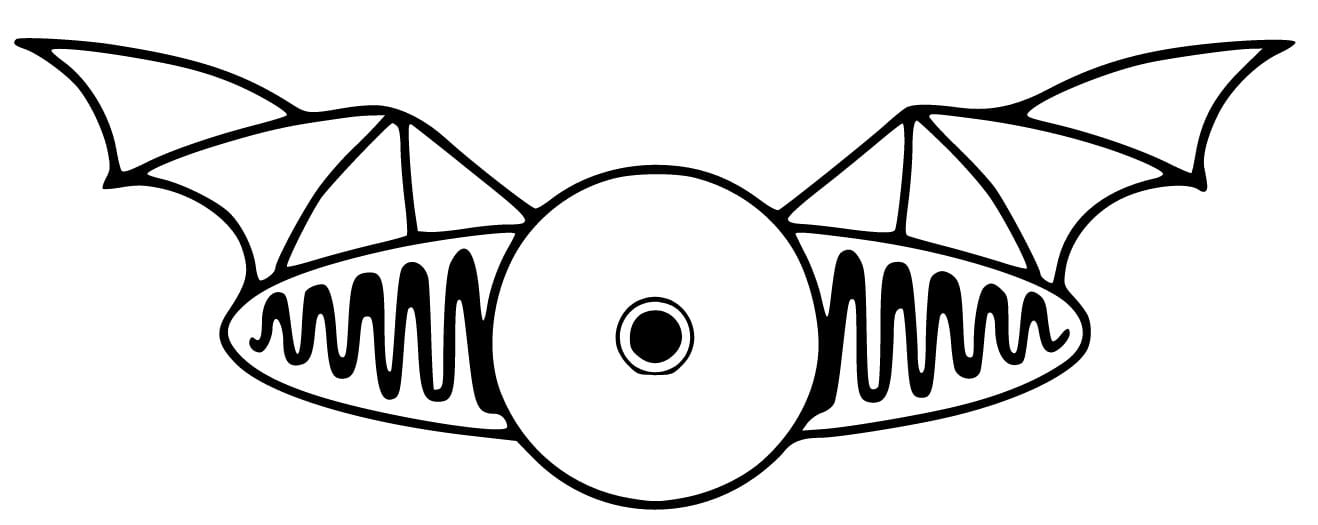
תרגילים אלו יכינו אתכם לקראת הקורס. הסגל מצפה מכם לסיים את כולם ולשלוח למייל הסגלי לבדיקה:

[segelmamas57@gmail.com](mailto:segelmamas57@gmail.com)

שימו לב להשקיע בתרגילים וללמוד מהם כמה שיותר – על השפה עצמה, על האלגוריתמים [והקונבנציות](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions). **ככל שתגדילו ראש לעשות זאת, יהיה לכם יותר קל בשלבים הראשונים של הקורס.** **לטובתכם!**

ההגשה היא דרך github. יש ליצור משתמש, ליצור repository ציבורי ולהעלות לשם את כל הקבצים הרלוונטיים [[סרטון tutorial]](https://www.youtube.com/watch?v=tRZGeaHPoaw&ab_channel=KevinStratvert) [[המדריך הרשמי של github]](https://docs.github.com/en/get-started/start-your-journey). כל חלק יהיה בתיקייה נפרדת משלו ויכיל את הקבצים הרלוונטיים לאותו חלק

**בהצלחה.**

****

שימו לב – אין להפיץ את השאלות לגורמים אזרחיים חיצוניים **בפקודה!**

# חלק ראשון – יכולות הקלדה

כדי לוודא שאתם יעילים מול המחשב, עליכם לעבוד על מהירות ההקלדה שלכם.

**מטרה סגלית מינימלית – 35 מילים לדקה (WPM)**

הסגל מציע 3 דרכים להתאמן:

* להיכנס לאתר [typing.com](https://www.typing.com/), ליצור משתמש תלמיד (חינמי) ולעבור שם את הקורסים הקשורים להקלדה [[ראו פה](https://www.typing.com/student/lesson/333/skill-builder)].
* במידה ואתם יודעים להקליד אבל לא מספיק מהר, עד תחילת הקורס להקדיש כל יום בין רבע שעה לחצי שעה להתאמן על יכולות ההקלדה שלכם. אתרים לדוגמה:
  + <https://play.typeracer.com/>
  + <https://monkeytype.com/>
* היכנסו למשחק [https://zty.pe](https://zty.pe/) והתחילו לשחק עד שאתם מרגישים שאתם שולטים ביכולות ההקלדה שלכם. לאחר מכן, לכו לאחד מהאתרים typeracer או monkeytype ובחנו את מהירות ההקלדה שלכם.

במידה ונכנסתם לאחד מאתרי ההקלדה הנ"ל והצלחתם להשיג את המטרה הסגלית המינימלית, המשיכו הלאה למשימות הבאות. במידה ואתם מתעכבים בחלק זה, שימו לב להתחיל לעבוד גם על החלקים האחרים במקביל כדי להספיק אותם.

**יש להוסיף לריפו התשובות שלכם screenshot של התוצאה.**

# חלק שני – אלגוריתמיקה

דגשים:

* את חלק זה יש לעשות בפייתון.
* יש לשמור על טייפים – גם עבור ערך ההחזרה.
* יש לשמור על קונבנציות PEP8 (נאכף ב-pycharm) וקוד נקי.
* כל שאלה צריכה להיות בקובץ נפרד.
* יש לפתור את התרגילים באופן הקצר ביותר והברור ביותר (הכירו את השפה כמה שיותר כדי לעשות זאת).
* הימנעו משימוש ב-chatGPT או כל מודל שפה כזה או אחר.
* הסגל מעריך פתרונות יצירתיים מחוץ לקופסה.

1. כתבו פונקציה בשם num\_len המקבלת מספר טבעי (אין צורך לבדוק תקינות קלט) ומחזירה את האורך שלו.
   * אין להשתמש במחרוזות בכלל.
   * אורך מימוש הפונקציה – שורה אחת בלבד.
   * רמז – היעזרו ביכולות המתמטיות שלכם מהתיכון.
2. שלשה פתגורית היא שלשה של מספרים טבעיים המקיימת את השוויון:

כתבו פונקציה בשם pythagorean\_triplet\_by\_sum אשר מקבלת מספר חיובי טבעי sum ומדפיסה את כל השלשות הפתגוריות שעונות על התנאי הבא:

שימו לב- יש להדפיס את השלשות בפורמט a < b < c.

1. כתבו פונקציה בשם is\_sorted\_polyndrom המקבלת מחרוזת ומחזירה true אם הוא פולינדרומי ממויין, אחרת false.

הגדרה לפולינדרומי ממויין – עד אמצע המחרוזת , כל האותיות\מספרים ממויינים באופן לקסיקוני

דוגמה לפולינדרומי לא ממוין – שוש (הסבר – האות ש' באה אחרי האות ו' ולכן מדובר בלא-ממויין)

דוגמה לפולינדרומי ממויין – "אבגדגבא" (הסבר- עד אמצע המחרוזת ומשני צידיה – האותיות מסודרות על פי סדר אלפבתי)

1. כתבו תכנית אשר קולטת מהמשתמש סדרת מספרים (לא בהכרח שלמים), עד אשר מתקבל הקלט 1-. כאשר כל המספרים בידיכם, עליכם לחשב ולהציג למשתמש:
   * ממוצע של מספרים.
   * כמות המספרים החיוביים.
   * מיון של המספרים בסדר עולה.
   * (בונוס) השתמשו ב-matplotlib וצרו גרף נקודות של הנתונים על פי סדר הגעתם.
   * (בונוס) חשבו את [מתאם פירסון](https://he.wikipedia.org/wiki/%D7%9E%D7%AA%D7%90%D7%9D_%D7%A4%D7%99%D7%A8%D7%A1%D7%95%D7%9F) של הנקודות שהתקבלו.
2. (בונוס) כתבו פונקציה בשם reverse\_n\_pi\_digits המקבלת מספר טבעי n (אין צורך לבדוק תקינות), ומחזירה את n הספרות הראשונות של פאי בסדר הפוך. ערך ההחזרה של הפונקציה הינו מחרוזת. אורך מימוש פונקציה – 3 שורות.

# חלק שלישי – תכנות מונחה עצמים

דגשים:

* את חלק זה יש לעשות ב-#C.
* יש ליצור פרויקט #C ולהגיש אותו עם כל הקבצים הנלווים.
* לרשותכם [קובץ PDF](https://did.li/PAJTY) המסביר את כל נושא תכנות מונחה עצמים, היעזרו בו.
* יש לשמור על [קונבנציות](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions) מתאימות וקוד נקי.
* כל שאלה צריכה להיות בקובץ נפרד. שאלות תיאורטיות יש להגיש בקובץ וורד אחד.
* יש לפתור את התרגילים באופן הברור ביותר (הכירו את השפה כמה שיותר כדי לעשות זאת).
* אין להשתמש ב-chatGPT או כל מודל שפה כזה או אחר.

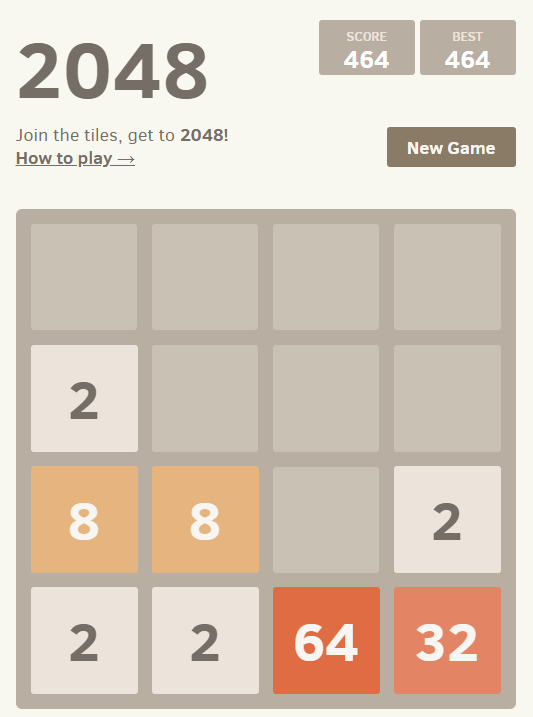
1. הסבירו מהו עקרון הכימוס.
2. מה ההבדל בין מחלקה למחלקה אבסטרקטית?
3. מה ההבדל בין מחלקה למחלקה סטטית?
4. מי יכול לגשת לmembers בעלי access modifier של internal ?
5. צרו מחלקה בשם Node שתכיל:
   1. Property בשם **Value** מסוג int המייצג את הערך שה-node מחזיק.
   2. Property בשם **Next** מסוג Node המייצג את ה-node הבא המחובר אליו.
6. צרו מחלקה בשם LinkedList שתכיל:
   1. משתנה פנימי מסוג Node (מהשאלה הקודמת) המייצג את ראש הרשימה המקושרת.
   2. פונקציה בשם **Append** המקבלת ערך מספרי ומוסיפה אותו כ-Node לסוף הרשימה המקושרת (אמור להיות (1)O)
   3. פונקציה בשם **Prepend** המקבלת ערך מספרי ומוסיפה אותו כ-Node לתחילת הרשימה המקושרת
   4. פונקציה בשם **Pop** שמורידה Node מסוף הרשימה ומחזירה את הערך שלו.
   5. פונקציה בשם **Unqueue** שמורידה Node מתחילת הרשימה ומחזירה את הערך שלו.
   6. פונקציה בשם **ToList** שלא מקבלת פרמטרים ומחזירה [IEnumerable](https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.ienumerable-1?view=net-6.0)<int> של הערכים של הרשימה המקושרת.
   7. פונקציה בשם **IsCircular** שלא מקבלת פרמטרים ומחזירה אם הרשימה המקושרת הנוכחית היא מעגלית או לא (בוליאני).
   8. כתבו פונקציה בשם **Sort** שעורכת את הרשימה המקושרת כך שהנודים יהיו ממויינים מהערך הקטן לגדול.
   9. פונקציה בשם **GetMaxNode** שתחזיר את ה-Node שמחזיק את הערך המספרי הגדול ביותר (אמור להיות (1)O).
   10. פונקציה בשם **GetMinNode** שתחזיר את ה-Node שמחזיק את הערך המספרי הקטן ביותר (אמור להיות (1)O).
7. צרו מחלקה בשם NumericalExpression:
   1. המחלקה תקבל בבנאי מספר
   2. המימוש שלה ל-[**ToString**](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/how-to-override-the-tostring-method) יהיה להחזיר את הביטוי המילולי של המספר. לדוגמה: עבור המספר 548 הפונקציה תחזיר את המחרוזת "Five Hundred Forty Eight"
   3. המספר המקסימלי אותה המחלקה יכולה לבטא הוא 999,999,999,999.
   4. המחלקה תכיל פונקציה בשם **GetValue** שתחזיר את המספר שהיא קיבלה בבנאי.
   5. המחלקה תכיל פונקציה סטטית בשם **SumLetters** המקבלת מספר ומחזירה את כמות האותיות שצריך כדי לכתוב את הביטויים המילוליים של כל המספרים מ-0 עד אותו מספר (בלי רווחים).
   6. צרו את **אותה הפונקציה** מהסעיף הקודם עם אותו שם בדיוק רק שתקבל במקום מספר מופע של NumericalExpression. איזה עקרון בתכנות מונחה עצמים בא לידי ביטוי? (יש לענות על התשובה הזו בהערה מעל הפתרון בקוד)
   7. (בונוס) כתבו בדרך כזו המאפשרת למשתמש להוסיף בצורה תכנותית עוד מימושים להמרות למילים כדי לאפשר:
      1. שפות שונות
      2. ביטויים מעבר ל999 מיליארד.

רמז: קראו על [Func](https://learn.microsoft.com/en-us/dotnet/api/system.func-2?view=net-8.0)

# (בונוס) חלק רביעי - 2048

דגשים:

* את חלק זה יש לעשות ב-C#
* יש לעשות אותו רק אחרי סיום חלק 2 ו-3.
* יש ליצור פרויקט #C נפרד ולהגיש אותו בגיטהאב עם כל הקבצים הנלווים. כל מחלקה,[Enum](https://www.w3schools.com/cs/cs_enums.php) , ממשק או [record](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/record) – בקובץ נפרד.
* יש לשמור על [קונבנציות](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions) מתאימות וקוד נקי. הימנעו מחזרתיות של קוד ([DRY](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself)) ושמרו על פשטות ([KISS](https://en.wikipedia.org/wiki/KISS_principle)).
* הקפידו על תיקיות וסדר בקבצים.
* הימנעו משימוש ב-[chatGPT](https://www.youtube.com/watch?v=dQw4w9WgXcQ&ab_channel=RickAstley) או כל מודל שפה כזה או אחר.



**משחק לדוגמה:** [**https://play2048.co**](https://play2048.co)

**הלוח:** המשחק מתבצע על לוח בגודל 4X4, כלומר 16 תאים.

**התחלת המשחק:** בתחילת המשחק, שני תאים מתמלאים באופן אקראי עם המספרים 2 או 4.

**המטרה:** להגיע לתא שערכו 2048.

**תזוזות:** השחקן בוחר להזיז את כל האריחים באחת מארבע הכיוונים: למעלה, למטה, ימינה או שמאלה. כאשר תאים עם אותו המספר נמצאים צמודים בכיוון התזוזה, הם יתמזגו לתא אחד עם הסכום של שני המספרים. (תשחקו במשחק המקורי כדי לראות כיצד התאים מתמזגים ולבדוק מקרי קצה)

**אריחים חדשים:** לאחר כל תזוזה, אריח חדש עם המספר 2 או 4 מופיע באופן אקראי באחד התאים הריקים בלוח.

**סוף המשחק:** המשחק נגמר כאשר אין תזוזות אפשריות נותרות והלוח מלא, או כאשר מגיעים לתא שערכו 2048.

**חישוב נקודות:** במשחק 2048, הנקודות מחושבות על פי ערך האריחים שמתמזגים. בכל פעם ששני אריחים עם אותו המספר מתמזגים לאריח אחד, תוספת הנקודות שהשחקן מקבל היא שווה לערך האריח החדש שנוצר מהמיזוג. לדוגמה, אם שני אריחים עם המספר 4 מתמזגים, הם יוצרים אריח עם המספר 8 והשחקן מקבל 8 נקודות.

להלן דרישות מקדימות, אלו החלקים בקוד **שדרוש** מכם לעשות. **כל דבר** מעבר לכך תשתמשו ביצירתיות שלכם והוסיפו בעצמכם. שימו לב לשמור על Naming ברור ותקין, וכן על קונבנציות #C תקינות.

**דרישות מקדימות:**

* צרו Enum בשם Direction המייצג את הכיוונים השונים שאפשר להזיז את התאים בלוח.
* צרו Enum בשם GameStatus המייצג את המצב של המשחק ויכיל 3 מצבים – Win, Lose, Idle
* צרו מחלקה בשם Board המייצגת את מבנה הנתונים של המשחק ומה ניתן לעשות עם הערכים. מחלקה זו תכיל:
  + Property בשם Data המייצג מערך דו מימדי של התאים (ה-Getter שלו צריך להיות ציבורי אבל ה-Setter צריך להיות protected – ראו [Properties](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/properties)).
  + פונקציה (שם לבחירתכם) האחראית על ההשמה הראשונית של 2 תאים במקומות רנדומליים עם הערכים 2 או 4.
  + פונקציה בשם Move המקבלת פרמטר מסוג Direction ותנסה להצמיד את התאים כמה שאפשר לכיוון שצויין. פונקציה זו גם אחראית על המיזוג ביניהם, והוספת תא נוסף עם הערך 2 או 4 במקום פנוי לבחירתה. ערך ההחזרה של הפונקציה יהיה כמה נקודות השחקן הרוויח באותו מהלך.
  + הרגישו חופשי להוסיף עוד פונקציות פרטיות כדי לפצל את הלוגיקה ולעשות את הקוד ברור יותר.
* צרו מחלקה בשם Game המכילה:
  + Property של Board המייצג את הלוח של המשחק.
  + Property של GameStatus המייצג את המצב של המשחק.
  + Property בשם Points המייצג את כמות הנקודות שהשחקן צבר (ה-Getter שלו צריך להיות ציבורי אבל ה-Setter צריך להיות protected – ראו [Properties](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/properties)).
  + פונקציה בשם Move המקבלת פרמטר מסוג Direction:
    - אם מנסים לקרוא לפונקציה כאשר GameStatus הוא Lose, הפונקציה לא תעשה כלום.
    - הפונקציה תנסה להצמיד לאותו כיוון את התאים.
    - לאחר הזזה, יש לבדוק את הסטטוס של המשחק.
    - לאחר הזזה, יש להוסיף ל-Points את כמות הנקודות שהמשתמש השיג בהזזה.
    - הפונקציה לא מחזירה כלום.
  + הרגישו חופשי להוסיף עוד פונקציות.
* צרו מחלקה בשם ConsoleGame האחראית על הצגת המשחק בקונסולה, קבלת קלט מהמשתמש דרך הקונסולה (דרך Console.ReadKey), והדפסת הודעות מתאימות למשתמש (עבור קלט לא תקין, ניצחון, הפסד).
  + אופן המימוש של המחלקה – לבחירתכם. יצירתיות!