

# תרגיל מס' 1 – Assemblies ושפת C#

## חלק 1 – גשש בלש

### מטרות

- הטמעת המושגים הבסיסיים בעולם הפיתוח עם ה- .Net Framework. כגון:
  - PE
  - Assembly
  - MSIL
  - Metadata
- הכרת הכלי ildasm שמהווה חלק מסט הכלים שמגיעים עם ה- .NET Framework.
- חשיפה לקוד MSIL
- חשיפה ל- Assembly Manifest
- הבנת היתרונות והחסרונות שבקוד מנוהל

### ידע נדרש

- הכרת המבנה של Assembly והמושגים השונים
- הכרת תחביר בסיסי בשפה הנתמכת ע"י ה- CLR (כדוגמת C#) כגון:
  - מחלקות - Classes
  - מרחב שמות – namespace
  - מתודות סטטיות
  - פרמטרים למתודות
  - משתני ערך (int, bool)
  - מחלקת Console
- עבודה עם הכלי – ildasm (Intermediate Language Disassembler)

### הכינותי מראש

- קובץ ה- zip (שמכיל את המסמך הזה) מכיל גם קובץ בשם Ex01.exe
- מחשב שמותקן עליו .NET Framework. לצורך הרצת התוכנית והרצת הכלי ildasm.

### התרגיל

נתון לכם קובץ Ex01.exe שהוא אפליקציה דוט-נטית.

בעת הרצת הקובץ תתבקשו להכניס שם משתמש וסיסמא על מנת להמשיך בפעולת התוכנית .

את שם המשתמש והסיסמא תגלו אחרי פעולות בילוש קצרות על הקובץ בעזרת הכלי לניתוח assemblies

שנקרא ildasm

תזכורת:

יש להריץ חלון console בסביבת דוט-נט (Developer Command Prompt for Visual Studio).

כדי להריץ אותו, יש לחפש אותו בעזרת תפריט החיפוש של Windows

(לפתוח את החיפוש ולכתוב Developer Command – הוראות מלאות ניתן למצוא [כאן](#))

ב- Command Prompt יש להגיע לתיקייה בה נמצא הקובץ Ex01.exe, ואז לכתוב את השורה הבאה –

```
[the dir of the exe file]>ildasm Ex01.exe
```

המשך בעמוד הבא..

**יש להפעיל ildasm על הקובץ ולענות על השאלות הבאות:**

1. האם הקובץ הזה הוא .NET Assembly? כן / לא (יש לסמן את התשובה הנכונה)

האם קובץ זה מהווה .NET PE? כן / לא (יש לסמן את התשובה הנכונה)

נימוק:

---



---



---



---



---

**תארו את ה- Assembly:**

- a. מה שמו? \_\_\_\_\_
- b. מה גרסתו? \_\_\_\_\_
- c. באיזה חלק של ה- assembly ישנן התשובות לסעיפים a, b? \_\_\_\_\_
- d. באילו assemblies אחרים הוא משתמש? (יש לתת שם וגרסה של כל אחד מהם)
- שם: \_\_\_\_\_ גרסא: \_\_\_\_\_
- שם: \_\_\_\_\_ גרסא: \_\_\_\_\_
- שם: \_\_\_\_\_ גרסא: \_\_\_\_\_

**2. ניתוח ה- MSIL**

עבור כל אחד מהטיפוסים בקוד, יש לתת תיאור מלא:

- a. סוג הטיפוס (class, struct, enum)
- b. שם הטיפוס
- c. רשימת הרכיבים (members) המוגדרים בטיפוס ותיאורם. התיאור כולל:
- i. Name
- ii. Type
- iii. Static / Instance
- iv. Access Modifier (public/private/protected)
- בנוסף, אם הרכיב הוא מתודה (פונקציה):
- v. סוג הערך המוחזר (return value type)
- vi. רשימת הפרמטרים שהיא מקבלת ותיאורם

**יש למלא טבלה כזו:**

Type (Struct/Class/Enum)	Name	Members (methods, fields)
Class	SomeClass	<ul style="list-style-type: none"> <li>public static int SomeMethod(float)</li> <li>...</li> </ul>
		<ul style="list-style-type: none"> <li></li> </ul>

3. מהם שם המשתמש והסימא הנדרשים להרצת התוכנית? (כאן נדרשת עבודת בילוש קטנה...)

4. הריצו את התוכנית ותוכלו להבין חלק מהמצופה בחלק מהתרגילים הבאים.

## חלק 2 – תכנות בסיסי

### מטרות

- פיתוח אפליקציות דוט נט בעזרת ה- Visual Studio
- תרגול בכתובת תוכנית בתחביר C#
- עבודה עם קלט/פלט בסביבת Console
- היכרות עם המחלקות string, int, float, char, math, StringBuilder

### ידע נדרש

- היכרות עם כלי הפיתוח Microsoft Visual Studio
- היכרות עם אלמנטים בסיסיים ותחביר בסיסי של שפת C#

### הכינותי מראש

- Microsoft Visual Studio מותקן על המחשב.

### התרגיל

יש ליצור Solution חדש **לפי השם שמוגדר בהוראות ההגשה (!!!)**.  
שימו לב, כדי ליצור קודם כל Solution ריק ורק אח"כ להוסיף לו פרויקטים:

File → New → Project → Blank Solution

עבור כל אחד מהסעיפים הבאים, יש ליצור פרוייקט נפרד בתוך ה- Solution.

[Right-Click on the solution node in the tree] → Add → New → Project... → Visual C# → Windows → Empty Project

שמות הפרויקטים יהיו בפורמט Ex01\_0X (כאשר X הוא מספר הסעיף).

- **להלן קישור למסמך המתאר באופן מפורט כיצד יש להכין את הפרויקטים לטובת התרגיל הזה - יש לעקוב אחר ההוראות בו באדיקות!**

**1. סדרות בינאריות**

כתבו תוכנית הקולטת מהמשתמש 3 מספרים בפורמט בינארי בני 8 ספרות כל אחד. אחרי כל מספר, המשתמש מקיש enter. (קלט לא חוקי ידווח למשתמש כלא חוקי והתוכנית תמתין לקלט חוקי). התוכנית תמיר את המספרים שהוכנסו למספרים עשרוניים ותדפיס אותם בפורמט עשרוני בסדר עולה. בנוסף, התוכנית תדפיס סטטיסטיקה לגבי הקלט:

- הערך הממוצע של שלושת המספרים בייצוג עשרוני
- רצף הביטים הארוך ביותר (אפסים או אחדות) מבין שלושת המספרים
- מספר החילופים (מעברים מ-0 ל-1 או מ-1 ל-0) בכל אחד מהמספרים
- המספר שיש לו את מספר ה-0 הגבוה ביותר ואת מספר ה-1 הנמוך ביותר

**יש לספק 5 דוגמאות הרצה כולל על הקלטים הבאים:**

א. 10101010, 11110000, 10000011  
מספרים עשרוניים בסדר עולה: 131, 170, 240  
ממוצע: 180.33  
רצף הביטים הארוך ביותר: 5 (10000011)  
מספר החילופים: 7 (10101010), 1 (11110000), 2 (10000011)  
המספר עם הכי הרבה 0 והכי פחות 1: 131 (בינארי: 10000011)

ב. 01010101, 11000111, 11111111  
מספרים עשרוניים: 85, 199, 255  
ממוצע: 179.67  
רצף הביטים הארוך ביותר: 8 (מהמספר השלישי, 11111111)  
מספר החילופים: 7 (01010101), 2 (11000111), 0 (11111111)  
המספר עם הכי הרבה 0 והכי פחות 1: 85 (בינארי: 01010101)

ג. 10011001, 11101111, 00000000  
מספרים עשרוניים: 0, 153, 239  
ממוצע: 130.67  
רצף הביטים הארוך ביותר: 8 (מהמספר השלישי, 00000000)  
מספר החילופים: 4 (10011001), 1 (11101111), 0 (00000000)  
המספר עם הכי הרבה 0 והכי פחות 1: 0 (בינארי: 00000000)

**2. עץ ה-ABC למתחילים**

כתבו תוכנית שמדפיסה למסך "עץ אותיות" באופן הבא (כולל מספרי השורות):

```

1      A
2      B C D
3      E F G H I
4      J K L M N O P
5      Q R S T U V W X Y
6      |Z|
7      |Z|

```

**בנוס: מימוש בעזרת רקורסיה.****3. עץ ABC דינאמי**

כתבו תוכנית שתעבוד כמו תוכנית עץ למתחילים, אך תקבל את גובה העץ המבוקש (כמות השורות הכוללת כולל הגזע) כקלט מהמשתמש (יש להעיר למשתמש על קלט לא תקין!).

**בנוס:** לממש זאת ע"י הפנייה ל- assembly שיצרתם בסעיף 2 (עץ ה- ABC למתחילים) ע"י כך שתוסיפו Reference מהפרויקט של הסעיף הזה לפרויקט של הסעיף הקודם.

\* גובה עץ מקסימלי הוא 15

**יש לספק דוגמאות הרצה (צילומי מסך) על הקלטים: 3, 4, 7, 9**

להלן דוגמא של עץ בגובה 5:

```

1      A
2      B C D
3      E F G H I
4      |J|
5      |K|

```

להלן דוגמא של עץ בגובה 9:

```

1      A
2      B C D
3      E F G H I
4      J K L M N O P
5      Q R S T U V W X Y
6      Z A B C D E F G H I J
7      K L M N O P Q R S T U V W
8      |X|
9      |X|

```

**4. ניתוח מחרוזת**

כתבו תוכנית המקבלת כקלט מחרוזת בגודל של 10 תווים (למשל: abccddccba) ומספקת את הפרטים הבאים:

1. האם המחרוזת היא **פלינדרום** (למימוש בעזרת רקורסיה)
  2. אם מדובר במספר - האם הוא מתחלק ב-4 ללא שארית..
  3. אם מדובר במחרוזת באנגלית – את מספר האותיות הקטנות שיש בה (lowercase)
  4. אם מדובר במחרוזת באנגלית - האם היא מסודרת בסדר אלפביתי יורד
- שימו לב, המחרוזת חייבת להיות מורכבת מאותיות באנגלית בלבד (גדולות או קטנות), או ממספרים בלבד.

מחרוזות כגון ab12cd34ef אינן חוקיות. שימו לב, **אותיות גדולות נחשבות שונות מאותיות קטנות.**

**יש לספק 6 דוגמאות הרצה כולל על הקלטים הבאים:**

**קלט "zyxwvutsrq" :**

1. האם פלינדרום: לא
2. מספר אותיות קטנות: 10
3. האם בסדר אלפביתי יורד: כן

**קלט "1234567890" :**

1. האם פלינדרום: לא
2. האם מתחלק ב-4 ללא שארית: לא

**קלט "aabbccbbaa" :**

1. האם פלינדרום: כן
2. מספר אותיות קטנות: 10
3. האם בסדר אלפביתי יורד: לא

**קלט "ZYxyZYxyZY" :**

1. האם פלינדרום: לא
2. מספר אותיות קטנות: 4
3. האם בסדר אלפביתי יורד: לא

**5. סטטיסטיקות מספרים**

כתבו תוכנית המקבלת מהמשתמש מספר שלם בן 9 ספרות ומספקת את הפרטים הבאים:

- א. כמה מהספרות גדולות מספרת האחדות במספר
- ב. כמה מהספרות במספר מתחלקות ב-4 ללא שארית
- ג. היחס בין הספרה הגדולה ביותר לספרה הקטנה ביותר (אם הספרה הקטנה היא 0, יש להתייחס לספרה הקטנה ביותר שאינה 0)
- ד. מספר הזוגות של ספרות זהות במספר

**יש לספק 5 דוגמאות הרצה כולל על הקלטים הבאים:**

**קלט: 000123456 (שימו לב הקלט תקין).**

ספרת האחדות היא 6. אפס ספרות גדולות ממנה. שתי ספרות מתחלקות ב-4 (0,4). היחס בין הספרה הגדולה ביותר (6) לקטנה ביותר שאינה 0 (1) הוא 6. יש 3 זוגות של ספרות זהות (שלושה אפסים).

**קלט: 112233445**

- א. מספר הספרות הגדולות מספרת האחדות (5): 0
- ב. מספר הספרות המתחלקות ב-4: 2 (4,4)
- ג. היחס בין הספרה הגדולה ביותר לספרה הקטנה ביותר: 5 (5/1)
- ד. מספר הזוגות של ספרות זהות: 4 (1-1, 2-2, 3-3, 4-4)

**קלט: 999888777**

- א. מספר הספרות הגדולות מספרת האחדות (7): 6
- ב. מספר הספרות המתחלקות ב-4: 0
- ג. היחס בין הספרה הגדולה ביותר לספרה הקטנה ביותר: 1.28 (9/7)
- ד. מספר הזוגות של ספרות זהות: 12 (3 זוגות של 9, 3 זוגות של 8, 3 זוגות של 7)

**שימו לב:**

1. **חובה** להשתמש במחלקות ובמתודות הבאות:
  - `StringBuilder` – טרם הכרנו. דורש מחקר עצמאי.
  - `string.Format`
  - `int.TryParse` – טרם הכרנו. יש דוגמא בדוגמת הקוד באתר.
  - `Math` – טרם הכרנו. דורש מחקר עצמאי.
  - `Char` – טרם הכרנו. דורש מחקר עצמאי.
  - מתודות נוספות של המחלקה `string` שיכולות לעזור.
2. את ההמרה של מספר בינארי לעשרוני יש לממש בעצמכם, ולא להשתמש במתודה קיימת שמבצעת זאת.
3. יש לוודא תקינות קלט בכל אחת מהמשימות, ולהציג למשתמש הודעה במקרה של קלט לא תקין ולאפשר לו להזין קלט תקין.
4. יש לספק דוגמאות הרצה לכל אחת מהמשימות.

**הגשה**

- ההגשה תתבצע לתיבת המייל של התרגילים של הקורס כמפורט במסמך הוראות הגשה שניתן למצוא באתר הקורס. **נקודות ירדו למי שלא יפעל בדיוק לפי ההוראות.**
- **אין צורך** לתעד בפורמט XML (עוד לא למדנו)
- **דוגמאות הרצה יש לספק ע"י הדבקת תצלומי מסך של ה-console למסמך וורד אותו אתם מגישים כפיתרון לחלקו הראשון של התרגיל. על קובץ ה docx. להימצא באותה תיקייה של קובץ ה- .sln שמו של קובץ: Ex01\_ScreenShots.docx**
- יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס.
- כל יום איחור יוריד 4 נקודות מציון התרגיל
- נא להימנע מהעתקות (הן מתגלות מאוד בקלות)