

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:

* בחלק מה MANIFEST השם של הקובץ מצוין כתור
assembly B25_Ex01 (בסורה 12) שם הוא .NET Assembly
* הקובץ מהווה .NET PE. השם של class ב B2025.Ex01.Program
יש מתודה בשם Main שבה יש entrypoint כמחזורי קובץ כר
הרצה (PE).

תארו את ה- Assembly:

- a. מה שמו? assembly B25_Ex01
b. מה גרסתו? 25:2:1:28312
c. באיזה חלק של ה- assembly ישנן התשובות לסעיפים a, b? MANIFEST
d. באילו assemblies אחרים הוא משתמש? (יש לתת שם וגרסה של כל אחד מהם)
שם: mscorlib גרסא: 4:0:0:0
שם: System.Xml גרסא: 4:0:0:0
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"> public static int SomeMethod(float) ...
		<ul style="list-style-type: none">

3. מהם שם המשתמש והסימא הנדרשים להרצת התוכנית? (כאן נדרשת עבודת בילוש קטנה...)
4. הריצו את התוכנית ותוכלו להבין חלק מהמצופה בחלק מהתרגילים הבאים.

Type (Struct/Class/Enum)	Name	Members (methods, fields)
Class	B2025.Ex01.Program	<ul style="list-style-type: none"> • public static void Main() • public ctor()
Class	B2025.Ex01.DemosExecutor	<ul style="list-style-type: none"> • public static void ShowMenu() • private static void ShowMenuOptions() • private static enum (eProgram Samples) getUserSelection() • private static bool authenticateUserCredentials() • public ctor()
		Enum eProgramSamples
		<u>Class DemoApps</u> <ul style="list-style-type: none"> • public ctor() • public static void RunBinarySeries() • public static void RunSandClockProgram() • private static int buildLine (class [mscorlib]System.Text.StringBuilder i-SB, int PlaceInText, int i-Rows) • private static bool checkIfAscendingSeries(String i-NumberStr) • private static bool checkIfDescendingSeries(String i-NumberStr) • private static void drawSandClock (int i-NumOfLines)

		<ul style="list-style-type: none">• private static string getInputFromUser(out int o_CurrentNumber)
--	--	---

DotNet: enjen de (3

C# rocks!! : 1<N0'0

חלק 2 – תכנות בסיסי

מטרות

- פיתוח אפליקציות דוט נט בעזרת ה- Visual Studio
- תרגול בכתיבת 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 4 מספרים בפורמט בינארי בני 7 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 יורד. בנוסף, התוכנית תדפיס סטטיסטיקה לגבי הקלט:

- הערך הממוצע של ארבעת המספרים בייצוג עשרוני
- רצף ה-1'ים הארוך ביותר מבין ארבעת המספרים (יש לציין את אורך הרצף ודוגמה למספר בו הוא מופיע).
- מספר החילופים (מעברים מ-0 ל-1 או מ-1 ל-0) בכל אחד מהמספרים
- המספר שיש לו את מספר ה-1'ים הגבוה ביותר
- מספר כולל של ביטים '1' שהופיעו בכל ארבעת הקלטים יחד.

יש לספק 4 דוגמאות הרצה כולל על הקלטים הבאים:

- א. 1111111, 0001110, 1110000, 1010101
- מספרים עשרוניים בסדר יורד: 127, 112, 85, 14
 - ממוצע: 84.5
 - רצף ה-1'ים הארוך ביותר: 7 (מהמספר 1111111)
 - מספר החילופים: 6 (1010101), 1 (1110000), 1 (0001110), 0 (1111111)
 - המספר עם הכי הרבה 1-ים: 127 (בינארי: 1111111)
 - מספר כולל של 1-ים: 17
- ב. 1011011, 0111110, 1000001, 0000000
- מספרים עשרוניים בסדר יורד: 91, 65, 62, 0
 - ממוצע: 54.5
 - רצף ה-1'ים הארוך ביותר: 5 (מהמספר 0111110)
 - מספר החילופים: 0 (0000000), 2 (1000001), 1 (0111110), 4 (1011011)
 - המספר עם הכי הרבה 1-ים: 91 (בינארי: 1011011) (או 62, שניהם עם 5)
 - מספר כולל של 1-ים: 12
- ג. 0011001, 1100110, 1010101, 0101010
- מספרים עשרוניים בסדר יורד: 102, 85, 42, 25
 - ממוצע: 63.5
 - רצף ה-1'ים הארוך ביותר: 2 (למשל מהמספר 1100110)
 - מספר החילופים: 6 (0101010), 6 (1010101), 3 (1100110), 3 (0011001)
 - המספר עם הכי הרבה 1-ים: 102 (בינארי: 1100110) (או 85, שניהם עם 4)
 - מספר כולל של 1-ים: 14

ד. דוגמת הרצה נוספת שלכם

2. עץ מספרים למתחילים

כתבו תוכנית שמדפיסה למסך "עץ מספרים" באופן הבא (כולל "מיספור" השורות עם אותיות):

```
A          1
B          2 3 4
C          5 6 7 8 9
D          1 2 3 4 5 6 7
E          8 9 1 2 3 4 5 6 7
F          |8|
G          |8|
```

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

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

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

* גובה עץ מקסימלי הוא 15, גובה עץ מינימלי הוא 4

יש לספק דוגמאות הרצה (צילומי מסך) על הקלטים: 4, 8, 9

להלן דוגמא של עץ בגובה 5:

```
A          1
B          2 3 4
C          5 6 7 8 9
D          |1|
E          |1|
```

להלן דוגמא של עץ בגובה 9:

```
A          1
B          2 3 4
C          5 6 7 8 9
D          1 2 3 4 5 6 7
E          8 9 1 2 3 4 5 6 7
F          8 9 1 2 3 4 5 6 7 8 9
G          1 2 3 4 5 6 7 8 9 1 2 3 4
H          |5|
I          |5|
```


4. ניתוח מחרוזת

כתבו תוכנית המקבלת כקלט מחרוזת בגודל של 12 תווים (למשל: bAax14aBccdd) ומספקת את הפרטים הבאים:

1. האם המחרוזת היא פלינדרום (Case Insensitive) (בעזרת רקורסיה)
2. אם המחרוזת מורכבת כולה מספרות:
האם המספר שהיא מייצגת מתחלק ב-3 ללא שארית?
3. אם המחרוזת מורכבת כולה מאותיות באנגלית:
כמה אותיות גדולות (UPPERCASE) יש בה?
האם התווים מסודרים בסדר אלפביתי עולה?

יש לספק 5 דוגמאות הרצה, כולל:

1. קלט aBcDeFgHiJkL :
 - האם פלינדרום: לא
 - מספר אותיות גדולות: 6
 - האם בסדר אלפביתי עולה: כן
2. קלט 123123123123 :
 - האם פלינדרום: לא
 - האם מתחלק ב-3 ללא שארית: כן
3. קלט aBCCBaABCCBa :
 - פלט צפוי:
 - האם פלינדרום: כן
 - מספר אותיות גדולות: 6
 - האם בסדר אלפביתי עולה: לא

- יש להוסיף עוד 2 דוגמאות הרצה לבחירתכם, המדגימות מקרים נוספים (למשל, קלט לא חוקי, פלינדרום מספרי, מחרוזת שאינה בסדר עולה וכו').

5. סטטיסטיקות מספרים

- כתבו תוכנית המקבלת מהמשתמש מספר שלם בן 8 ספרות ומספקת את הפרטים הבאים:
- א. כמה מהספרות קטנות מהספרה הראשונה (הכי שמאלית)
 - ב. כמה מהספרות במספר מתחלקות ב-3 ללא שארית
 - ג. ההפרש בין הספרה הגדולה ביותר לספרה הקטנה ביותר
 - ד. מהי הספרה המופיעה מספר הפעמים הרב ביותר במספר, וכמה פעמים היא מופיעה?

[דוגמאות הרצה בעמוד הבא]

יש לספק 4 דוגמאות הרצה כולל:

1. קלט 31415926 :
 - ספרה שמאלית: 3. ספרות קטנות ממנה (לא כולל הראשונה): 1, 1, 2. סה"כ: 3 :
 - ספרות המתחלקות ב-3: 3, 9, 6. סה"כ: 3 :
 - הפרש: 8
 - הספרה השכיחה ביותר: 1 (מופיע פעמיים)
 2. קלט 00770088 :
 - ספרה שמאלית: 0. ספרות קטנות ממנה (לא כולל הראשונה): אין. סה"כ: 0 :
 - ספרות המתחלקות ב-3: 0, 0, 0, 0. סה"כ: 4 :
 - הפרש $8-0=8$
 - הספרה השכיחה ביותר: 0 (מופיעה 4 פעמים)
- יש להוסיף עוד 2 דוגמאות הרצה לבחירתכם, המדגימות מקרים נוספים.
 - **אסור להשתמש במערכים / מבני נתונים**

שימו לב:

1. **חובה** להשתמש במחלקות ובמתודות הבאות:
 - `StringBuilder` – לא הכרנו. דורש מחקר עצמאי.
 - `string.Format`
 - `int.TryParse` – לא הכרנו בשיעור, נכיר בהמשך. יש דוגמא בדוגמת הקוד באתר.
 - `Math` – לא הכרנו בשיעור. דורש מחקר עצמאי.
 - `Char` – לא הכרנו בשיעור. 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.**

הגשה

- ההגשה תתבצע לתיבת המייל של התרגילים של הקורס כמפורט במסמך הוראות הגשה.
- 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.doc
- יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי.
- כל יום איחור יוריד 4 נקודות מציון התרגיל.
- נא להימנע מהעתקות (הן מתגלות מאוד בקלות)

בהצלחה ☺