**שיעורי בית 4: זיכרון דינמי**

**הנחיות**

* יש להגיש את שיעורי הבית דרך מערכת הבדיקה האוטומטית ודרך מערכת ה-neo, כקובץ מכווץ הנושא את שמכם ומס' העבודה. לדוגמה: IdanEx4.zip.
* כל תרגיל צריך להופיע כפרויקט נפרד באותו קובץ .sln
* בכל ()main צריך להיות ()getchar בסוף התוכנית.
* במקרים בהם ישנה דוגמת ריצה – חקו אותה במדויק.
* ודאו שכל הזיכרון שהקצתם משוחרר ושאין בקוד זליגות זיכרון!

**שאלות**

1. בשאלה זו נתנסה במחקר זליגות זיכרון:

א) כתבו תוכנית שמקצה זיכרון באופן דינמי ל-int אחד ומסיימת את ריצתה מבלי לשחרר את הזיכרון. הריצו את הכלי לבדיקת זליגות זיכרון על תוכנית זו וצרפו תמונת מסך שמפרטת את הדליפה.

ב) כעת, תקנו את הקוד כך שהזיכרון ישוחרר. הריצו שוב וצרפו תמונה שמאשרת שכל הזיכרון שוחרר.

ג) כתבו תוכנה שמקצה זיכרון באופן דינמי עפ"י מספר הנקלט מהמשתמש, ומסיימת את ריצתה מבלי לשחרר את הזיכרון. הריצו את הכלי לבדיקת זליגות זיכרון על תוכנית זו וצרפו תמונת מסך שמפרטת את הדליפה.

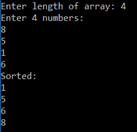
ד) כעת, תקנו את הקוד כך שהזיכרון ישוחרר. הריצו שוב וצרפו תמונה שמאשרת שכל הזיכרון שוחרר.

**הדרכה:** עבור שאלה זו עליכם להגיש את קבצי הקוד של סעיפים ב' ו-ד' במערכת האוטומטית וארבעה צילומי מסך (אחד עבור כל סעיף).

2. בשאלה זו נתרגל הקצאה דינמית של מערך.

* קלטו מהמשתמש מספר שלם, וצרו **מערך דינמי** בגודל הערך שנקלט.
* קלטו לתוכו מספרים מהמשתמש.
* כתבו פונקציה הממיינת את המערך שיצרתם ומדפיסה אותו (כל ערך בשורה נפרדת). הפונקציה מקבלת כפרמטרים מצביע למערך ואת גודלו.   
  **הדרכה:** היעזרו באפשרויות למיון משיעורים 1 + 2.
* אל תשכחו לשחרר את כל הזיכרון שהוקצה דינמית כאשר סיימתם להשתמש בו.

**דוגמת ריצה:**



3. שאלת מחקר:

הקצאה דינאמית ב-heap מאפשרת לנו גם לשנות גודל של מערכים בזמן ריצה.

חקרו על הפונקציה realloc וענו על השאלות הבאות:

א. מה תפקידה של הפונקציה?

ב. איזה פרמטרים היא מקבלת (התייחסו גם לתפקיד וגם לטיפוס)?

ג. מהו ערך ההחזרה של הפונקציה?

ד. מדוע לדעתכם כדאי להשתמש ב-realloc?

4. נכתוב מערכת פשוטה לניהול ציונים שמשתמשת בהקצאה דינמית ומאפשרת גמישות רבה!

המחשבון ישאל את המשתמש כמה תרגילים היו עד כה הסמסטר – ויקצה דינמית מערך מספרים באורך זה. לאחר מכן יקרא המחשבון מהמשתמש את כל הציונים שלו (מספר בטווח 0-100 – ודאו תקינות).

למשתמש ינתנו האפשרויות הבאות:

1 – חישוב ממוצע

2 - שינוי אחד הציונים. המשתמש יישאל איזה ציון ברצונו לשנות (מספר בין 1 למספר הציונים שהוכנס – ודאו תקינות הקלט), ולאחר מכן למה הוא רוצה לשנותו (ערך חדש בטווח 0-100. שוב – ודאו תקינות).

3 – שינוי מספר הציונים. במידה והמערך גדל – המשתמש יתבקש להכניס את הציונים החסרים.

4 – הדפסת הציונים לפי הסדר.

5 – יציאה.

**שימו לב:**

המחשבון מוודא שהציון הוא מספר בטווח 0-100 – בצעו בדיקה זו באמצעות פונקציה. אין לשכפל קוד!

לשינוי מספר הציונים השתמשו ב-realloc.

**דוגמת ריצה:** ראו קובץ מצורף gradesCalc.exe.

5. אחרי שהצליח להתגבר על כל קשיי הפויינטרים, דניאל החניך המתוסכל ממשיך לא להתייאש!

וגם עכשיו, הוא יגיש את הקוד הכי טוב שיש!

לשם כך הוא מבקש מחברו, רון אלוף הזיכרון, לעבור על הקוד שכתב בהדרגה,

ולאשר שאכן לא הייתה בו שום זליגה!

חקרו את הקובץ leaksFree.c.

א) האם רון מצא זליגות זיכרון? אילו שגיאות קיימות בקוד? פרטו כהערה בקובץ.

ב) תקנו את כל השגיאות שמצאתם.

6. **בונוס:** כיצד פונקציית free יודעת כמה זיכרון עליה לשחרר? (הרי היא מקבלת רק מצביע לזיכרון שהוקצה).

7. **בונוס יצירתי:** חשבו על שמות לשאר חניכי הכיתה של דניאל החניך המתוסכל.

כל שם צריך להתחרז עם שם תואר רלוונטי לתכונות של מגשימיסטים.

**בהצלחה!**