**תכנות מונחה עצמים שיעור 1**

**קישור לגיט:** https://github.com/AhuvaElkarif/OOP\_Course/blob/main/OOP/Program.cs

**תרגיל 2**

|  |  |  |
| --- | --- | --- |
| **גודל מערך מקומי (byte)** | **משתנים מקומיים נוספים** | **קריאות** |
| 100 | ללא | 9557 |
| 100 | משתנה אחד (int) | 8694 |
| 100 | שני משתנים (int) | 8689 |
| 1000 | ללא | 9620 |
| 1000 | משתנה אחד (int) | 8742 |
| 1000 | שני משתנים (int) | 8692 |
| 10000 | משתנה אחד (int) | 8684 |

**תובנות:**

1. ככל שגודל המערך המקומי גדל, מספר הקריאות המקסימלי קטן.
2. הוספת משתנים מקומיים מפחיתה את עומק הקריאות המקסימלי, משום שהם צורכים מקום בזיכרון ה- stack.

**תרגיל 3**

**ניתוח תוצאות**

**א. הקצאת מערכים מסוגים שונים**

* הקצאת מערך דורשת זיכרון עבור המבנה הבסיסי של המערך: אורך, מצביעים וכדומה.
* לדוגמה:
  + מערך של int דורש זיכרון בסיסי, בתוספת הזיכרון של כל איבר.
  + מערך של string כולל רק הפניות, והאובייקטים עצמם מוקצים בזיכרון אחר.

**ב. הקצאת מערך של Struct**

* Struct מוקצה כולו בזיכרון המחסנית (stack).
* הזיכרון שהוקצה תלוי ישירות בגודל ה-Struct:
  + Struct קטן דורש פחות זיכרון פר איבר.
  + Struct גדול יגרום למערך לצרוך יותר זיכרון.

**ג. הקצאת מערך של Class**

* מערך של Class כולל רק הפניות בזיכרון, ולכן הזיכרון הבסיסי של המערך קטן יחסית.
* גודל הזיכרון בפועל תלוי בגודל כל אובייקט שמוקצה בנפרד על ה-Heap.
* לדוגמה:
  + Class קטן יגרום לזיכרון להיות קטן פר אובייקט.
  + Class גדול יגרום לזיכרון לצרוך יותר פר אובייקט.

**מסקנות**

* **זיכרון בסיסי למערך:** כל מערך דורש כמות זיכרון בסיסית עבור אורך ומצביעים.
* **Struct:** הקצאת זיכרון תלויה בגודל המבנה (struct).
* **Class:** הזיכרון למערך הוא עבור הפניות בלבד, אך זיכרון נוסף מוקצה על ה-Heap לכל אובייקט.

**תרגיל 4**

**מה התוכנית מנסה לעשות?**

התוכנית מנסה להרחיב מערך a כך שאורכו יוכפל, והנתונים המקוריים שבו יועתקו פעמיים במערך המורחב.  
לדוגמה, אם המערך המקורי הוא {1, 2, 3}, המערך המורחב אמור להיות {1, 2, 3, 1, 2, 3}.

**מדוע זה לא מצליח?**

התוכנית לא מצליחה לבצע את ההרחבה מכיוון שהשינוי שבוצע במערך בתוך הפונקציה ExpandArray לא משפיע על המשתנה a שבתוכנית.

הסיבה לכך היא שבשפת C# מערכים מועברים לפונקציות לפי **הפניה**, אך **המשתנה עצמו (המצביע למערך) מועבר לפי ערך**.  
במילים אחרות:

* הפונקציה ExpandArray מקבלת עותק של המצביע למערך.
* כאשר הפונקציה יוצרת מערך חדש (array = new int[...]), היא משנה רק את העותק של המצביע, ולא את המצביע המקורי שבתוכנית.

**כיצד ניתן לתקן את התוכנית?**

**פתרון 1: החזרת המערך המורחב לפונקציה הקוראת**

אפשר לתקן את התוכנית כך שהפונקציה ExpandArray תחזיר את המערך המורחב, והקריאה לפונקציה תעדכן את המשתנה a בהתאם.

**פתרון 2: שימוש במשתנה ref**

ניתן להשתמש במילה השמורה ref כדי להעביר את המשתנה לפי הפניה אמיתית, כך שהשינוי שבוצע בפונקציה ישפיע ישירות על המשתנה המקורי.