

שאלה 1 – קובץ נפרד בגיט (P1Q1)

שאלה 2 – הערך של `a[0]` ישתנה בקוד הנתון כי שליחת מערך לפונקציה לא שולחת העתק של המערך (בניגוד למשתנה של ערך בודד) אלא שולחת את המיקום של התא הראשון בזכרון ואת מספר התאים במערך. לכן שינויי שקורה לאותו המערך בפונקציה אחרת יגרר גם לפונקציה הקדומה (MAIN) במקרה זה

שאלה 3 – בניגוד לשאלה קודמת כאן ערכו של `a[0]` לא ישתנה בפונקציה `main` מאחר ולמרות שאנו שולחים את המערך לפונקציה הפעולה ראשונה שעשינו הייתה ליצור מערך חדש עם הפקודה: `a = new int[3];` וזה למעשה העביר את מיקום המערך למקום חדש בזכרון ולא השפיע על המערך המקורי. בנוסף המערך החדש לא הוחזר ל-`main` ולכן לא שינה את המערך המקורי שנוצר ב-`main`.

שאלה 4 – קובץ נפרד בגיט (P1Q4)

שאלה 5 – נשתמש בפרמטר המוגדר על ידי `params` עבור מערך שאיננו יודעים כמה משתנים אמורים להיכנס לתוכו. למשל אם נרצה ליצור פונקציה שתחזיר סכום של מספר לא ידוע של משתנים בכל פעם כלומר:

```
static void Main(string[] args)
{
    int sum1 = SumFunc(1, 2);
    int sum2 = SumFunc(1, 2, 3, 4);
}

private static int SumFunc(params int[] values)
{
    int sum = 0;
    for (int counter = 0; counter < values.Length; counter++)
    {
        sum = sum + values[counter];
    }
    return sum;
}
```

בקוד הנ"ל המשתנה `sum1` מבקש מהפונקציה סכום של שני ערכים בעוד `sum2` מבקש סכום של ארבע משתנים אך שניהם פונים לאותה הפונקציה.

אתגר: השימוש בparams בדו"כ יעשה עבור סיפריות שימשו מספר רב של תוכניות (כמו למשל SumFunc בדוגמא למעלה) אך שימוש זה גוזל הרבה שטח זכרון ביחס לשימוש במערך רגיל בעל גודל ידוע ולכן לפעמים עדיף להשתמש במערך גדול במיוחד על פני params בתוכניות רגישות לגודל זכרון.

שאלה 6 – קובץ נפרד בגיט (P1Q6)