**שיעורי בית 3: מערך כמצביע**

**הנחיות**

* יש להגיש את שיעורי הבית דרך מערכת הבדיקה האוטומטית ודרך מערכת ה-neo, כקובץ מכווץ הנושא את שמכם ומס' העבודה. לדוגמה: YaelEx3.zip.
* כל תרגיל צריך להופיע כפרויקט נפרד באותו קובץ .sln
* בכל main צריך להיות getchar בסוף התוכנית.
* יש לוודא קימפול וריצה תקינה לכל פתרון שאתם מגישים!
* אלא אם נאמר אחרת – ספציפית בתרגיל זה אין להשתמש באופרטור [ ] (אופרטור לקבלת איבר במערך), מלבד לשם אתחול מערכים. אנחנו רוצים לתרגל עבודה ישירה עם מצביעים, כתובות וזיכרון.
* במקרים בהם ישנה דוגמת ריצה - יש לחקות אותה במדויק.
* על שאלות פתוחות ענו כהערה בקובץ הקוד המתאים.

**שאלות**

1. כתבו שתי פונקציות המקבלות מערך מספרים שלמים ואת גודלו.

הפונקציה arrayInput תקלוט איברים למערך והפונקציה arrayReverseOutput תדפיס את איברי המערך למסך בסדר הפוך מסדר הכנסתם.

כתבו main מתאים המאתחל מערך בגודל 10, קורא לפונקציה הראשונה, ואז לשנייה.

תזכורת: אין להשתמש באופרטור [ ].

דוגמת ריצה:



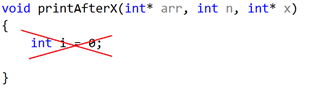
2. כתבו פונקציה אשר מקבלת מערך, את גודלו, ומצביע לאיבר מסוים בזיכרון.

תיאור: הפונקציה תבדוק האם המצביע מצביע לאחד מאיברי המערך.

כלומר, נמצא בטווח הכתובות של המערך:

* אם כן, הפונקציה תדפיס את כל איברי המערך שנמצאים לאחר הכתובת שנשלחה- לא כולל המצביע עצמו.
* אם לא, הפונקציה תדפיס הודעה מתאימה.

דגשים:

* אין להגדיר משתנים מקומיים בפונקציה, אלא להשתמש אך ורק בפרמטרים שנשלחים! (תזכורת: ראינו בשיעור שגם פוינטר יכול להיות משתנה לולאה).  
  
* אתחלו את המערך בערכים הבאים:

arr[] = {4 ,8 ,6 ,2 ,1 ,3 ,5 ,7 ,8 ,9 ,5};

* אין להשתמש ב-[ ] בעת גישה לאיברי המערך.

בפונקציית ה-main עליכם לקלוט מהמשתמש מספר שלם, לחבר אותו אל תחילת המערך ולשלוח את התוצאה כמצביע לפונקציה (כפי שתואר בתיאור הפונקציה).

המספר שתקלטו ישמש כהיסט מתחילת המערך (ראו תמונת "קריאה לפונקציה").

חתימת הפונקציה:

void printAfterX(int\* arr, int n, int\* x);

קריאה לפונקציה:

printAfterX(arr, size, arr + offset);

דוגמאות ריצה:





במילים פשוטות, לאחר הקלט של מספר שלם, יש לשלוח פונקציה את המצביע למערך+המספר השלם, ולבדוק את הכתובת הזו נמצאת בטווח "האינדקסים".

3. בשאלה זו נכתוב פונקציה המקבלת טקסט מוצפן ומפענחת אותו.

הפונקציה תעבוד כך: היא תקבל מחרוזת (אותיות קטנות, אורך עד 50 תווים) ומפתח (מספר שלם) n ובסיום ריצתה באותה המחרוזת יהיה הטקסט המפוענח.

הוראות הפענוח:

א) ראשית, יש להפוך את הטקסט ככתב מראה. למשל, המילה olleh תהפוך למילה hello.

ב) לאחר הפיכת הטקסט, כל אות מוחלפת באות הנמצאת n מקומות אחריה לפי הא'-ב'

(בצורה מעגלית - אחרי ת' תבוא א').

* עבור n=3: האות a תוחלף באות d
* עבור n=9: האות b תוחלף באות k
* עבור n=1: האות z תוחלף באות a

חישבו על דרכים חכמות לכתוב את הפונקציה ולא להשתמש ב-for/if מסובכים (רמז, ולא בפעם הראשונה: מודולו %).

כתבו פונקציית main המקבלת צופן לפענוח, ומפתח, ומפענחת אותו.

חתימת הפונקציה:

void decrypt(char\* cipher, int n);

תזכורת:

אין להשתמש באופרטור [ ] .

דוגמת ריצה:



4. דניאל החניך המתוסכל הבין שהשם של מערך הוא בעצם פוינטר. הוא מתרגש, ורוצה להשתמש בידע החדש על מנת להדפיס הודעה חשובה. הביטו בקובץ myMsg.c, וענו על השאלות:

א) מהי שגיאת זמן הריצה שנפלה בקוד? מדוע היא מתרחשת? הסבירו זאת בהערה בקוד.

ב) תקנו את השגיאה.

ג) בונוס: עם מי דניאל רוצה להיפגש?

5. **שאלת מחקר:** כתבו פונקציה אשר מייצרת מערך ומחזירה את כתובתו.

לאחר מכן הדפיסו את המערך באמצעות פונקציה אחרת.

האם הצלחתם להדפיס את איברי המערך? לדעתכם מדוע זה קרה?

6. **שאלת מחקר - באג בסיכון אבטחתי:** בשאלה זו נעסוק בפרצת האבטחה buffer overflow. כזכור למדנו עליה קצת בסמסטר א'.

הביטו בקובץ secretFiles.c.

התכנית קוראת סיסמה מהמשתמש ומשווה אותה לסיסמה השמורה בתור קבוע בקוד.

במידה והסיסמה נכונה התכנית מאפשרת למשתמש גישה לקבצים הסודיים מאד!

בתכנית זו קיימת שגיאת buffer overflow.

ב-VS קיימים שירותים שמונעים ניצול לרעה ב-buffer overflow. לכן, על מנת שתרגיל זה יעבוד, עליכם לקמפל ולהריץ אותו בעזרת gcc בחלון הCMD (כמו שעשינו בסמסטר א') ולא בvisual studio:

א) הריצו את התכנית. נסו להכניס את הסיסמא הנכונה ולראות איך התכנית מגיבה. אחר כך הכניסו סיסמא שגויה (עד 16 תווים) והתבוננו איך התכנית מגיבה.

ב) כעת הכניסו סיסמא ארוכה מ-16 תווים. (נסו סיסמא של לפחות 30 תווים). מה קרה כעת בתכנית? הבחינו שאנחנו משתמשים ב-scanf לקליטת המחרוזת!

ג) תקנו את השגיאה. סגרו את פרצת האבטחה. הצילו את הקבצים הסודיים.

7. **בונוס מגניב:** כתבו פונקציה המקבלת שני מצביעים: אחד לתחילת מערך בגודל 11 ואחד למיקום כלשהו במערך.

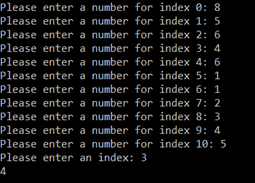
על הפונקציה להחזיר את מס' האיברים מתחילת המערך ועד לאינדקס הנבחר (כולל).

השתמשו ב- main שכתבתם לשאלה 2 על מנת להגדיר ולאתחל את המערך, לקלוט אינדקס מהמשתמש ולשלוח מצביעים לפונקציה.

הגבלות:

* אין להשתמש בלולאות.
* אורך הפונקציה צריך להיות **שורה אחת בדיוק**.

דוגמת ריצה:



8. **בונוס העשרה: מה קורה בזיכרון?** בשאלה זו נבין קצת יותר את האופן בו זיכרון מוקצה למשתנים.

ב-main, הגדירו שלושה משתנים מסוג int, והכניסו לתוכם את המספרים 1,2,3.

לאחר מכן, הגדירו מערך intים ובו 3 מקומות, והכניסו לשם את הערכים 4 5 6.

הדפיסו את הכתובות של המשתנים לפי הסדר של הערכים שלהם – 1 עד 6. קודם כל את הכתובת של 1 ולבסוף את הכתובת של 6.

א) שרטטו את מבנה הזיכרון - טבלה ובה הכתובות (לפי הסדר) ואיזה ערך יש בכל כתובת.

ב) הכרזנו על המשתנים "לפי הסדר" – מ-1 ועד 6. האם הם הוקצו בזיכרון בסדר הזה? באיזה סדר הם הוקצו?

ג) כעת, כשאתם מבינים קצת יותר טוב מה קורה בזיכרון, חזרו לשאלה 6 בתרגיל זה והסבירו:

מה קרה בתכנית כאשר הכנסנו סיסמא ארוכה מ-16 תווים, ואיך זה גרם לתוצאה הלא צפויה.

ייתכן שהדפסת הכתובות תעזור לכם בשאלה זו.

**בהצלחה!**