

| סמסטר | מועד | סעיף | שאלה  | תשובה   |
|-------|------|------|---|---|
| 2017א | 84   | א    | התוכנית הראשית (main) חייבת להחזיר ערך לסביבת העבודה. זאת בכדי להעביר לסביבה סטטוס הצלחה/כישלון.  | לא נכון. פונקציית החזרה יכולה להיות פונקציית void.  |
|       |      | ב    | שדה סיביות (bit field) יכול להיות בעל 8 שדות לכל היותר.   | לא נכון   |
|       |      | ג    | תכניות המשתמשות באופן ישיר בקריאות מערכת ההפעלה הן ניידות (portable). כלומר, ניתן להריץ במערכת הפעלה אחרת, ללא התאמה למערכת, בדומה לשימוש בספריה הסטנדרטית.   | לפעמים  |
| 2016ב | 90   | א    | במקרה של מספרים מסוג integer, פעולת & (and ביטים) ופעולת && (and לוגי) הן למעשה שקולות.   | לפעמים נכון ולפעמים לא. פעולת & היא פעולת השוואה בין סיביות של שני מספרים, בעוד ש && זהו ביטוי לוגי שמחזיר 1 או 0. במקרים מסוימים כמו במספרים 1 ו-0 נקבל את אותה התוצאה בין שני הפעולות & &&. במקרים אחרים, לדוגמה שנווה בין המספרים 2, עבור הפעולה & נקבל שהתשובה היא 2 (כי זה משווים בין הביטים שדלוקים במקום השני) אבל עבור && נקבל 1 כלומר true ששני הערכים זהים. |
|       |      | ב    | כאשר משתנה אוטומטי (מקומי לפונקציה) מוגדר כ-static, הוא הופך למעשה למשתנה גלובלי.   | לא נכון. משתנה לוקלי שהוא סטטי אומנם נשמר ב data ולא במחשנית אך אין גישה אליו מחוץ לבלוק הפונקציה בה הוא מוגדר, ולכן אינו זהה למשתנה חיצוני.  |
|       | 87   | ג    | קובץ אשר נפתח על ידי fopen, לא נסגר על ידי מערכת ההפעלה, בסיום התכנית. לכן יש צורך בשימוש ב-fclose, לשם סגירתו.   | לא נכון. עמוד 162 בספר: fclose נקרא באופן אוטומטי בסיום כל תוכנית ע"י המערכת. עם זאת, מומלץ לסגור את הקובץ בסיום השימוש.  |
|       |      | א    | אם לאופרטור אריתמטי יש שני אופרנדים: אחד מסוג int והשני מסוג float, אזי האופרנד מטיפוס float יומר לטיפוס int.   | לא נכון, לפי עמוד 44 בספר, אם אחד ה data types הוא float אז type ה float הנמוך יותר יקודם להיות גם הוא מסוג float.  |
|       |      | ב    | הפונקציה main נקראת בתחילת התכנית, אולם ניתן לקרוא גם לה מתוך פונקציות אחרות, כפי שניתן לקרוא לכל פונקציה אחרת בתכנית.  | נכון, אם כי זה מקרה פחות נפוץ. ניתן לקרוא לה מתוך פונקציה אחרת לפי ההגדרות הרגילות של C.  |
| 2016א | 93   | א    | שימוש במצביע (pointer) לבלוק זיכרון שקול, למעשה, לשימוש במערך (array).  | לפעמים נכון ולפעמים לא. אין הגבלה לגבי מספר הקבצים שניתן לעדכן, אולם יש הגבלה על מספר הקבצים שתוכנית יכולה לפתוח במקביל ובנוסף ישנה הגבלה על מס' התוכניות שיכולות לפתוח את אותו הקובץ במקביל.   |
|       |      | ב    | לאחר שהוגדר אב הטיפוס של פונקציה (prototype), יש להעביר בקריאה לפונקציה, מספר ידוע של ארגומנטים, כמספר שנקבע בהגדרת אב הטיפוס.                                | נכון, אנו נקבל שגיאה אם לא נקרא לפונקציה בהתאם לגדרת האב טיפוס (החתימה של הפונקציה).  |
|       |      | ג    | הפורמט והתוכן של תיקיה (directory) זהים בכל מערכת הפעלה. לפיכך ניתן לכתוב פונקציית חיפוש לקובץ, בתוך תיקיה, ולהשתמש בפונקציה זו ב-UNIX ובכל מערכת הפעלה אחרת. |   |
|       | 86   | א    | בכל פעם שהמהדר מטפל במחרוזת קבועה, הוא מקצה מקום, בגודל השווה לאורך המחרוזת + 1, בחלק ה-data של התכנית.   | נכון, התו האחרון מצביע על סיום המחרוזת '0\'   |
|       |      | ב    | ערכו של משתנה a מסוג מערך (array) הוא הכתובת של האלמנט הראשון במערך a[0].   | נכון, הוא מצביע על האיבר הראשון ברשימה, עמוד 28.  |
|       |      | ג    | בכדי לקרוא בתים ניתן להשתמש ב-system call (קריאת מערכת) "read". מומלץ לקרוא את כל הבתים יחדיו, ולא לקרוא בית אחד בכל קריאה.                                   | נכון, כי זה יעיל יותר שיש פחות קריאות מערכת ברגע ששולחים כמה ביחד ולא אחד אחד. עמוד 170.  |
|       | 84   | א    | הקצאת זיכרון למשתנה חיצוני (גלובלי) נעשית כפעולה ראשונה, בריצת התכנית.  | לא נכון, הקצאת זיכרון למשתנים לא נעשית במסגרת ריצת התוכנית אלא בתהליכים הקודמים לה.   |
|       |      | ב    | הצבת הקבוע 0 לתוך מצביע, היא שגיאה, מכיוון ש-0, אינו מען חוקי עבור נתונים.  | לא נכון, הצבת הקבוע 0 זהה להצבת הקבוע NULL שערכו שווה ל-0.  |
|       |      | ג    | ניתן להשתמש במשתנה מסוג unsigned short בבטחה, מכיוון שגודלו זהה תמיד, בכל מערכת.  | לא נכון, תקן C-ANSI לא מחייב מה יהיה גודל של קבוע זה, ייתכן שיהיה שונות בין מערכות שונות, אך בפועל ברוב מערכות ההפעלה גודלו זהה.  |

|       |    |   |  |   |
|-------|----|---|--|---|
| 2015ב | 83 | א | אין חשיבות לסדר הפונקציות בתוכנית.   | לפעמים, במידה ולא מתבצע שינוי על משתנים המשותפים לכמה פונקציות. לדוגמה אם נקרא לפונקציות שמבצעות פעולות אריתמטיות נקבל הבדל. כמו ההבדל בסדר פעולות חשבון. |
|       |    | ב | בפקודה printf, המהדר מוודא שמספר הערכים המודפסים, זהה למספר המשתנים המתואר במחרוזת הפקודה.   | לא נכון. הקומפלייר רק מוודא שנשלח ערך כלשהו (יכול להיות גם NULL). אם לא ישלח ערך הוא ידפיס ערך כלשהו.   |
|       |    | ג | אין להשתמש בקריאת המערכת "creat" עבור קובץ קיים.   | לא נכון, עמוד 173 בספר כן ניתן להשתמש.  |
|       | 87 | א | ההחלטה איזה סוג של חילוק יתבצע, חילוק בין מספרים ממשיים או בין מספרים שלמים, מתבצעת על-פי ההקשר (context), שבו מופיע החילוק.   | נכון, לפי הסוג של המשתנים אותם מחלקים.  |
|       |    | ב | ניתן לקרוא, במפורש, את EOF לתוך משתנה, ללא שגיאת ריצה.   | לפעמים  |
|       |    | ג | במערכת UNIX ניתן לקרוא מקובץ, בסדר כלשהו, מבלי לקרוא סדרתית את כל התווים שבו.  | נכון  |
| 2015א | 83 | א | פתיחת קובץ לא קיים לקריאה (למשל על ידי fopen) תגרום לשגיאה בזמן הריצה.   | לא נכון, אם ארעה שגיאה תחזיר הפונקציה NULL.   |
|       |    | ב | אם מספר האיברים באתחול רשומה (structure), קטן ממספר השדות ברשומה, האתחול נעשה לשדות הראשונים, ואילו שאר שדות הרשומה, מקבלים ערכים לא מוגדרים.  | לא נכון, int יאותחלו ב0 ופונטרים יאותחלו בNULL.   |
|       |    | ג | במערכת UNIX יש לכתוב ממשק (interface) מיוחד ושונה לכל קלט ולכל פלט. למשל, הממשק עבור המקלדת יהיה שונה מהממשק עבור המסך.  |   |
| 2014ב | 84 | א | הקוד הבא:<br>int a = 0;<br>a++ = 5;<br>הוא קוד חוקי.   | לא נכון, צד שמאל (המשתנה) חייב להיות מסוג שמכיל value ולא פעולה אריתמטית.   |
|       |    | ב | ניתן לבצע אריתמטיקה של מצביעים. למשל, אפשר לבצע כפל ב-2 של מבציע למערך כדי להגיע לאיבר אחר במערך.  | לא נכון. ניתן לבצע פעולות חיבור אך לא פעולות כפל. *p = p+2; Char *p יעבור בניגוד ל *p = p*2; char נקבל כאן שגיאת קומפילציה.                               |
|       |    | ג | במערכת UNIX לא ניתן לקבוע מראש את הרשאות הקובץ. למשל, לא ניתן ליצור קובץ לקריאה בלבד, יש לייצרו ורק אחר-כך לשנות את ההרשאה.  |   |
| 2013ב | 85 | א | אם הפונקציה func מופיע באב טיפוס (prototype) באופן הבא:<br>double func(double);<br>אז בקריאה לפונקציה:<br>x = func(3);<br>המספר 3 (שהוא integer) יהפוך לערך 3.0 (שהוא double) ללא צורך בהמרה (cast). | נכון  |
|       |    | ב | כאשר רוצים לחסוך בזיכרון, ניתן להשתמש ב-bit-field. באופן זה אורזים כמה אובייקטים במילת זיכרון אחת. כך גם ניתן להתייחס לכתובת כל ביט, על ידי שימוש באופרטור הכתובת '&'.                               | לא נכון   |
|       |    | ג | כדי לקרוא בתים ניתן להשתמש ב-system call (קריאת מערכת) "read". מומלץ לקרוא בית אחת בכל קריאה.  | לא נכון, עדיף לקרוא כמה בתים בקריאה אחת כי זה יעיל יותר שיש פחות קריאות מערכת ברגע ששולחים כמה ביחד ולא אחד אחד. עמוד 170.                                |

| סמסטר | מועד | סעיף | שאלה  | תשובה   |
|-------|------|------|---|---|
| 2013א | 84   | א    | ערכו של משתנה a מסוג מערך (array) הוא הכתובת של האלמנט הראשון במערך a[0].   | נכון  |
|       |      | ב    | משתנים חיצוניים מוגדרים מחוץ לפונקציות, ולכן מוכרים על ידי כל הפונקציות בתכנית.   | לא נכון   |
|       |      | ג    | הקצאת זיכרון (storage allocation) אינה תלויה מכונה, כך שניתן להעביר תכניות C ממערכת UNIX למערכת אחרת, ללא חשש.  | לא נכון   |
|       | 81   | א    | כאשר משתמשים באופרטור מודולוס % (modulus) עבור משתנים מסוג float, נלקח החלק השלם של כל משתנה.<br>לדוגמה:<br>float x = 20.5;<br>float y = 4.0;<br>float z;<br>z = x%y;<br>בתוכנית זו יתבצע בעצם:<br>20%4 ונקבל את התוצאה 5 במשתנה z. | לא נכון   |
| 2012ב | 87   | ב    | משתנה שהוגדר כ-const זהה למעשה לערך קבוע, שהוגדר על ידי #define.  | לפעמים, ישנם const שניתנים לשינוי בעוד שdefine לא ניתן לשינוי בשופ צורה. ניתן לאתחל את הקבועים במספרים שונים אך לא את define. |
|       |      | ג    | לא ניתן לקרוא לקריאות מערכת ההפעלה UNIX, מתוך התכנית הראשית. יש לכתוב פונקציית ספריה מיוחדת, אשר תבצע את הקריאה.  | לא נכון   |
|       |      | א    | פונקציה מחזירה תמיד ערך, אלא אם כן מוצהר שהערך החוזר הוא void.  | נכון  |
| 2011ב | 82   | ב    | אם מספר האיברים באתחול רשומה (structure), קטן ממספר השדות ברשומה, האתחול נעשה לשדות הראשונים, ואילו שאר השדות ברשומה, מקבלים ערכים לא מוגדרים.  | לא נכון   |
|       |      | ג    | שימוש בקריאת המערכת lseek מאפשר שיטוט בקובץ. כתוצאה מהשימוש בה עלול להשתנות תוכן הקובץ.   | לא נכון   |
|       |      | א    | משתנה המוגדר מחוץ לפונקציה הוא למעשה גלובלי, ומוכר בכל מקום בתכנית.   | לא נכון   |
| 2010ב | 87   | ב    | ניתן להעביר רשומה כפרמטר לפונקציה רק לפי כתובת.   | לפעמים  |
|       |      | ג    | שימוש בקריאת המערכת lseek מאפשר שיטוט בקובץ. אך כתוצאה משימוש בה עלול להשתנות תוכן הקובץ.   | לא נכון   |
|       |      | א    | אם הפונקציה func מופיע באב טיפוס (prototype) באופן הבא:<br>double func(double);<br>אז בקריאה לפונקציה:<br>x = func(3);<br>המספר 3 (שהוא integer) יהפוך לערך 3.0 (שהוא double) ללא צורך בהמרה (cast).                                | נכון, שפונקציה אמורה לקבל double והיא מקבלת משתנה int, מתבצע קסטינג אוטומט ואין צורך לעשות קסטינג מפורש.                      |

| סמסטר | מועד | סעיף | שאלה  | תשובה  |
|-------|------|------|---|--|
|       |      | ב    | האופרטור '&' מבצע הקצאת זיכרון עבור משתנה אשר לא בוצעה עבורו הקצאה קודמת. לדוגמה, במקרה של הפקודה: <code>b = &amp;a</code> מוקצה מקומו של משתנה b בזיכרון, ומושווה למקומו של a. | לא נכון, האופרטור & מסמן שאנחנו רוצים לעשות השמה(להעביר) של הכתובת של משתנה a למשתנה b.  |
|       |      | ג    | במערכת UNIX ניתן לפתוח, בו-זמנית, קבצים ללא הגבלה, על ידי שימוש בפונקציית המערכת "open".  | לא נכון יש הגבלה על מספר הקבצים שניתן לפתוח (בספר כתוב לעיתים קרובות 20)   |
| 2010א | 82   | א    | לא ניתן להגדיר משתנה, בתוך בלוק פנימי של פונקציה (סוגריים מסולסלים פנימיים). ניתן להגדיר את כל המשתנים בתחילת הפונקציה בלבד.  | לא נכון, אם כי נהוג להגדיר משתנים בתחילת הפונקציה. עמוד 9 בספר פסקה ראשונה. ניתן להגדיר משתנה גם בתוך פונקציה כל עוד מגדירים אותו לפני השימוש.                                       |
|       |      | ב    | אין חשיבות לערכו של EOF, כל עוד הוא שונה מערכם של כל התווים האחרים במערכת.  | תמיד נכון. ערכו של EOF משמש לזיהוי סוף קובץ / קלט ולכן אם הוא שונה משאר התווים אין הבדל לערכו. עמוד 152 בספר שורה ראשונה.  |
|       |      | ג    | קובץ אשר נפתח על-ידי fopen, נסגר תמיד על-ידי מערכת ההפעלה, בסיום התוכנית. כך שאין צורך מעשי בשימוש ב-fclose.  | נכון אבל לא תמיד. הקובץ ייסגר בסיום התוכנית, אבל ישנן הגבלות על פתיחה של אותו קובץ ממספר תוכניות שונות, ולכן מומלץ לסגור בסיום השימוש על מנת לא לפגוע בשימוש של תוכניות אחרות בקובץ. |
| 2009ב | 92   | א    | האופרטור '&' מבצע הקצאת זיכרון עבור משתנה אשר לא בוצעה עבורו הקצאה קודמת. לדוגמה, במקרה של הפקודה: <code>b = %a</code> מוקצה מקומו של משתנה b בזיכרון, ומושווה למקומו של a.     | לא נכון, כשהאופרטור & לפני המשתנה זה גורם להשמת של כותבת המשתנה ולא הערך שלו, בדוגמא אנחנו עושים השמה לכתובת של a למשתנה b, מה שקורה בעצם, שהערך של b הוא הכתובת של a.               |
|       |      | ב    | הצבת הקבוע 0 לתוך מצביע, היא שגיאה, מכיוון ש-0, אינו מען חוקי עבור נתונים.  | לא נכון, הצבת 0 לתוך מצביע גורמת לכך שהערך שלו יהיה null, וזאת אינה שגיאה, במידה וננסה לקבל את הערך במוצבע נקבל שגיאה כי הוא null והוא לא מצביע לאף מקום.                            |
|       |      | ג    | תיקייה (Directory) שמערכת UNIX היא בעצם קובץ, ורשימת תכונותיה זהה לתכונות קובץ רגיל: שם, גודל, תאריך יצירה וכו'.  | נכון, במערכת הפעלה unix, כל תיקיה היא בעצם קובץ רגיל שמכיל את כל המידע על התיקיה.  |