

## מבחן סופי - קורס בדיקות תוכנה 67778 – סמסטר ב' 2019

מועד ב' (תאריך: 4 אוגוסט 2019)

מרצים: מיכאל שטאל, שמואל גרשון

אינטל, ירושלים

ורסיה זו כוללת את ההערות שאמרנו בכיתה בזמן המבחן

### הוראות:

בבחינה 10 שאלות. כל השאלות יבדקו. הניקוד לכל שאלה מופיע בסוגריים מרובעות ליד מספר השאלה. כתבו את התשובות במחברת הבחינה. אין צורך להחזיר את דפי הבחינה. זמן הבחינה – 180 דקות. מותר להשתמש בכל חומר עזר מלבד טלפונים, מחשבים או ציוד תקשורת אחר (גם לא דגלי סמפור). ניתן לרשום הערות על הבחינה או על שאלות ספציפיות, במחברת הבחינה.

בהצלחה!

## תרחיש א'

מדפסת משולבת מסוגלת לבצע את הפעולות הבאות:

- הדפסת מסמכים (print)
- סריקת מסמכים והעברתם למחשב (scan)
- שכפול מסמכים (copy)
- שליחת וקבלת פקס (fax)

כל אחת מהפעולות הן כמו שאתם מכירים בחייכם האמיתיים (כלומר, בבית ובמשרד).

ניתן לקנפג את המדפסת בכל רגע נתון דרך פונקציית "הגדרות" (Settings). ניתן להדליק או לכבות כל אחת מפעולות המדפסת כרצוננו.

המדפסת תוכננה עם ההתנהגויות הבאות:

גם כשפעולה הודלקה, היא תהיה פעילה רק אם הפעולות שמעליה, לפי סדר הרישום למעלה, הופעלו (למשל, על מנת שאפשר יהיה לשכפל מסמכים, חייבים קודם להפעיל את "הדפסת מסמכים" וגם את "סריקת מסמכים").

גם כאשר כל ארבעת הפעולות מופעלות בהגדרות, אפשר לשלוח פקס רק אם קו הטלפון מחובר.

סריקה והדפסה במדפסת יכולות להתבצע בשחור לבן או בצבע. סריקה או הדפסה בצבע מתאפשרת רק אם בדיוק שתיים מתוך ארבע הפעולות של המדפסת דלוקות בהגדרות. (הערה שנאמרה באולם הבחינה: במשפטים אלה, השימוש במילה "הדפסה" מתייחס גם לביצוע שיכפול, או קבלת פקס. גם במקרים אלה מתבצעת הדפסה של מסמך; כלומר, המשפטים מתייחסים לכל הפעולות האפשריות במדפסת, לא רק לשתיים הראשונות).

עליכם לבדוק שאכן הכללים שהוגדרו כאן יושמו נכון בתוכנה המשובצת במדפסת.

(1) [20]

בנו טבלת החלטות **מצומצמת** עבור ההגדרות והכללים שתוארו (כלומר: עם שימוש ב- "don't care" שיסומן ב- "~" במקומות שבהם זה מתאים).

(2) [10]

הרצתם את כל הטסטים על פי טבלת ההחלטות וכולם עברו בהצלחה. בכל זאת החלטתם לפתוח באג על השימושיות (Usability) של המדפסת. מה נראה לכם לא סביר בהתנהגות המדפסת בהשוואה למדפסות המוכרות לכם? כתבו את דוח הפגם (bug report).

(3) [10]

הרצת הבדיקות של המדפסת נעשות כרגע על ידי בודק שמפעיל את המדפסת, משנה את ההגדרות דרך מסך מגע ומוודא גם את התנהגות המדפסת וגם את תוצאות ההדפסה על נייר. בדיקות אלה צורכות זמן, נייר ודיו רבים. הציעו שתי יכולות עזר לבדיקה (Testability hooks) שאם ימומשו בקוד של המדפסת יאפשרו יעילות גבוהה יותר בבדיקות.

**הדרכה:** צריך לפתור שתי בעיות: איך לבצע את הגדרות ופעולות במדפסת ואיך לוודא שהתוצאות של פעולות המדפסת תקינות. מדובר על פתרון בתוכנה ללא תוספת חומרה (מצלמה, רגש וכו').

כתבו את הפתרון בצורה של דרישות נוספות מהמערכת. הדרישות יכולות לתאר חלק מפתרון מלא, אם הפתרון המלא מוסבר בתיאור כללי. לא נעמוד על קוצו של יוד בבדיקת הדרישות, אבל לא נקבל תיאור כללי ומעורפל שניתן להבין בדרכים רבות.

(4) [10]

אם משתמש מנסה להשתמש בפעולה שלא הודלקה (דרך "הגדרות"), המדפסת מציגה הודעת שגיאה מתאימה למסך. כתבו את מקרה הבדיקה (test case) עבור בדיקה זו. מקרה בדיקה זה יורץ באופן ידני.

(5) [10] [5 לכל סעיף]

(א) בדרישות למדפסת יש בעיה של אי-התאמה (inconsistency). מה היא?  
(ב) כתבו דרישה לא-פונקציונאלית למהירות ההדפסה של המדפסת.

סוף תרחיש א'

תרחיש ב'

פיתחתם משחק רשת לטלפונים ניידים. האפליקציה צריכה לעבוד היטב בתנאי העבודה הבאים:

- a. מערכות הפעלה: Android 6.x , Android 7.x
- b. גודל מסך טלפון: 1280×720 , 1334×750 , 1920×1080
- c. מודלים של טלפונים: Lenovo B5 ,Xiaomi Two ,Redmi 10Z ,Xperia SX ,Pixel 3M ,Samsung T7
- d. טכנולוגית תקשורת: GSM, LTE, CDMA, UMTS, HSDPA
- e. רשתות: 2G, 3G, 4G, 5G

(6) [5]

כל מקרה בדיקה של האפליקציה צריך להיות מורץ מספר פעמים, על מנת לכסות את תנאי העבודה השונים שפורטו לעיל.

כמה פעמים צריך להריץ כל בדיקה בכל אחת מהאסטרטגיות הבאות:

- a. Each Choice
- b. Base Choice
- c. All combinations

(7) [10] [5 לכל סעיף]

עליכם להחליט באיזה תנאי עבודה (כלומר: באיזה קומבינציות של טלפון, OS, מסך וכו') להשתמש עבור הפעילויות הבאות. עבור כל פעילות, רשמו שני שיקולים שיש להביא בחשבון ואיך שיקולים אלה יביאו להחלטה. לא צריך להציע ממש איזה קומבינציה לקחת; רק מה השיקולים בבחירתה וההסברים למה שיקולים אלה רלוונטיים לפעולת הבדיקות.

**הערה:** מותר להניח הנחות לגבי מאפיינים של המשחק שאתם מפתחים וצריכים לבדוק – אך יש לציין הנחות אלה.

פעילויות הבדיקה:

(א) בדיקות ביצועים (Performance).

(ב) בדיקות אוטומטיות המתבצעות כל לילה על הקוד, כחלק מ- continuous integration.

### המשך תרחיש ב'

הקוד למטה הוא חלק מתוכנת המשחק. השתמשו בו כדי לענות על שאלות 8 ו-9.

```
int SUCCESS = 0;
int ERROR = 99;

char username[32];

// Secure function that appends a random, 3 digit number to the end of
// the global variable "username".
void RandomizeUserName()
{
    // The code for this function is not important.
    // After executing this function, the size of the global
    // "username" content is always smaller than 25 chars and
    // is NULL terminated.
}

// Asks the player for their username, to set the global variable.
int GetUsername()
{
    char user_input[32];

    printf("Welcome, new player. Please enter your username.\n");
    scanf("%s", user_input);
    if (strlen(user_input) >= 20)
    {
        printf("Error adding user_name to game. Try again.");
        return ERROR;
    }
    strncpy(username, user_input, 32);
    RandomizeUserName();
    printf("Game on! Your username: ");
    printf(username);
    return SUCCESS;
}
```

[7] (8)

בבדיקת Security נמצא כי פונקציית `GetUsername()` פגיעה לתקיפות אבטחה. מה היא פירצת האבטחה? הסבירו איך ניתן לנצל את הפירצה ותנו דוגמה לקלט שינצל אותה. אין צורך לכתוב קוד.

[6] (9)

א) כתבו בדיקת יחידה (unit test) לפונקציית `GetUsername()` הבודקת ש `username` הגדול מ 20 תווים נדחה. (רמז: stub)

ב) כתבו בדיקת יחידה (unit test) לפונקציית `GetUsername()` הבודקת ש `username` נקלט כראוי (רמז: mock)

הערה: בתשובות ה-syntax המדויק ממש לא חשוב, pseudocode יספיק.

## סוף תרחיש ב'

(10) [12]

נתונות הפונקציות הבאות:

```
def foo(myList):  
    if len(myList) < 5:  
        return -1  
  
    a = myList[3]  
    b = myList[4]*myList[2]  
  
    ret = bar(a, b)  
  
    if ret == 0:  
        return 2*a  
  
    if ret == 1:  
        return 2*b  
  
def bar(a, b)  
    if (a < b):  
        a = a+b  
    if (a > b):  
        a = a-b  
    if (a == b) || ((a < 50) && (b > 100)): +  
        return 0  
    else  
        return 1
```

(א) הציעו סט מינימלי של טסטים ששיגי 100% כיסוי משפטים (statement coverage) עבור foo().  
(כלומר: רשמו את ערכי myList שאיתם נקרא ל-foo() בכל טסט).

(ב) הציעו סט מינימלי של טסטים ששיגי 100% כיסוי תנאים (condition coverage) עבור bar().  
(מספיק לציין סטים של ערכי a ו-b; לא צריך לציין את ערכי myList; הניחו שאין כלל short circuiting).

ג) בודקת טענה: "בלתי אפשרי להשיג 100% כיסוי משפטים עבור foo() ובאותה עת להשיג 50% כיסוי החלטות (decision coverage) עבור bar()".  
האם הבודקת צודקת? אם כן, תנו את סט הבדיקות המשיג מטרות זו. אם לא, הסבירו למה הטענה נכונה.

---

## תשובות

(1)

Conditions	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Print	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Scan	~	N	Y	N	N	N	Y	Y	Y	Y
Copy	~	N	N	Y	N	Y	N	Y	Y	Y
Fax	~	N	N	N	Y	Y	Y	N	Y	Y
Phone line	~	~	~	~	~	~	~	~	N	Y
Actions										
Print	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Scan	N	N	Y	N	N	N	Y	Y	Y	Y
Copy	N	N	N	N	N	N	N	Y	Y	Y
Fax	N	N	N	N	N	N	N	N	N	Y
Color printing	N	N	Y	Y	Y	N	N	N	N	N
B&W printing	N	Y	Y	Y	Y	Y	Y	Y	Y	Y

(2) ישנן שלוש בעיות שימוש עיקריות במדפסת.

- א) העובדה שבכלל צריך לקנפג איזה פעולה עובדת; זה לא נהוג במדפסות משולבות שיש בשוק
- ב) נניח שיש סיבה לצורך לקנפג (אולי: אני רוצה לחסום לילדי המדפיסים בלי חשבון את היכולת לשכפל דפים אבל רוצה שיוכלו להדפיס עבודות שקיבלו מבית הספר). ברור שהכל יכול לעבוד יחד – הרי אפשר להדליק את כל הפעולות – אז למה הדרישה לעשות זאת לפי סדר מסוים? ולמה דווקא בסדר הזה? לגמרי לא הגיוני למשתמש.
- ג) נניח שיש צורך לחסום הדפסה צבע (מילדי המדפיסים וכו'). למה הדרישה שהדפסת צבע עובדת רק כששתי פעולות דלוקות? זה מגביל בצורה לא הגיונית, ויחד עם הדרישה לאיפסור לפי סדר חוסם את האפשרות לשכפל בצבע.

כל תיאור באג שיכסה אחד או יותר מהבעיות שהוזכרו – מקובל.

הנה דוגמא שמכסה את כל שלושת הבעיות יחד.

**כותרת:** פעולת המדפסת וקביעת ההגדרות שלה אינן אינטואיטיביות, שונות מהמקובל בשוק ומגבילות את השימוש במדפסת ללא צורך.

[הערה: כיוון שכאן לא מדובר בבאג כתוצאה מפעולה ברורה, קצת קשה להכניס את הכותרת לתבנית שנלמדה, אם כי לא בלתי אפשרי:  
**Action:** פעולת המדפסת וקביעת ההגדרות שלה  
**Symptom:** אינן אינטואיטיביות, שונות מהמקובל בשוק ומגבילות את השימוש במדפסת ללא צורך.]

בבחינה נקבל כותרות שאינן מתאימות לתבנית, בתנאי שהן קריאות, מובנות ומעבירות נכון את תמצית הבעיה. ]

**תיאור:** מדפסות משולבות הקיימות בשוק מאפשרות שימוש בכל אחת מארבע הפעולות וכן הדפסה בצבע ללא דרישה להגדרה כלשהיא. הצורך של המשתמש להגדיר איזו פעולה מאפשרת, וכן ההגבלות של סדר האפשרויות הופכות את השימוש במדפסת ללא אינטואיטיבי.

בנוסף, הדרישה שהדפסה בצבע עובדת רק כששתי פעולות מאפשרות, בצירוף עם הדרישה שפעולה מופעלת רק כאשר הפעולות מעליה מופעלות, משמעותה היא שהדפסה בצבע אפשרית רק עבור פעולת ההדפסה, ולא עבור שכפול או הדפסת פקס.

[ניתנים כאן שתי אפשרויות לצעדים לשחזור – עבור האספקטים השונים של הבעיות במדפסת; מובן שבבחינה נסתפק בסט אחד לשחזור עבור אחת מהבעיות]

#### **צעדים לשחזור (הגדרת פעולות המדפסת):**

- (1) אפשר דרך "הגדרות" את פעולת ההדפסה והשכפול בלבד
- (2) נסה לשכפל מסמך

**תוצאה צפויה (על ידי משתמש סביר):** ניתן לשכפל מסמך

**תוצאה בפועל:** לא ניתן לשכפל מסמך

#### **צעדים לשחזור (עבור איפשרויות הדפסה בצבע):**

- (1) אפשר דרך "הגדרות" את כל הפעולות
- (2) שלח מסמך להדפסה בצבע
- (3) צלם מסמך צבעוני

**תוצאה צפויה (על ידי משתמש סביר):** הדפסה ושכפול בצבע אפשריים

**תוצאה בפועל:** הדפסה או שכפול בצבע אינה אפשרית בקונפיגורציה זו

**חומרה:** קריטית

**הסבר לחומרה המוצעת:** לדעתי אין סיכוי להצלחה מסחרים למדפסת אם הדרישות ישארו כמות שהם. גם מי שיקנה אותה בטעות יחזיר אותה מיד. כל המדפסות המשולבות האחרות בשוק לא מגבילות את המשתמש בשום צורה.

(3)

#### **ביצוע הגדרות ופעולות במדפסת:**

נוסף שני APIs למערכת שיאפשרו לשנות את ההגדרות ולהפעיל את המדפסת בצורה אוטומטית

**הנחה:** המדפסת מחוברת ל-PC דרך הרשת או דרך כבל. ניתן לקרוא ל-APIs ממחשב המחובר למדפסת.

API א': קינפוג המדפסת:

SetPrinterSettings(int print, int scan, int copy, int fax)

for each parameter:



0: disabled

1: enabled

API ב': הפעלת המדפסת:

הנחה:

- הדפסה מופעלת באופן אוטומטי על ידי שליחת קובץ מהמחשב
- סריקה, שכפול או שליחת פקס מתבצעים על מסמך שהונח מראש על הזכוכית של המדפסת.

Int ActivatePrinter(int action, string &amp;param1)

**Action:**

- |               |  |
|---------------|--|
| 0: do nothing | param1: ignored  |
| 1: scan       | param1: name of the file on the computer into which the scan will be saved |
| 2: copy       | param1: ignored  |
| 3: send fax   | param1: phone number   |

**Return value:**

- 0: Success
- 1: The action is not enabled. In this case, param1 will be updated with the error message sent to the printer's touch screen.

**וידוא שתוצאות ההדפסה טובות:**

נוסיף משתנה בהגדרות של המדפסת (ניתן להדליק אותו ידנית או דרך API). אם משתנה זה דלוק, המדפסת לא תדפיס ממש אלא תשלח את נתוני ההדפסה לקובץ על המחשב המחובר למדפסת. כיוון שאנו מכירים את המדפסת אנו יודעים איזה מידע צריך להשלח לחומרה על מנת להפעיל אותה כשורה (אפשר למשל להקליט פעם אחת מה הביטים הנשלחים לחומרה עבור הדפסת מסמך ידוע). לאחר ביצוע "הדפסה" כזאת, ניתן להשוות את מה שהתקבל בקובץ מהמדפסת למה שאחנו מצפים לראות (הקובץ המוקלט). ווריאציה ליישום כזה היא שהמדפסת תשלח את נתוני ההדפסה ל"מדפסת pdf" ותייצר קובץ pdf התואם למה שהתכוונה לשלוח לחומרה שמבצעת את ההדפסה.

בצורה כזאת ניתן לבדוק שהמדפסת תקינה גם ללא הדפסה ממש, ובאופן אוטומטי.

(4)

**כותרת:** הודעת שגיאה מתאימה כתוצאה משימוש בפעולה שאינה מאפשרת

**מטרה:** בדיקה זו מוודאת שנסיון שימוש בפעולה שאינה מאפשרת גורם להודעת השגיאה הנכונה על מסך המגע של המדפסת.

**מתודולוגיה:** בעזרת ההגדרות, מבטלים יכולת פעולה מסוימת ולאחר מכן מנסים להשתמש בה. פעולה זו תייצר את הודעת השגיאה הרלוונטית. לפני תחילת הבדיקה מוודאים שכל הפעולות – אם הן מאפשרות – אכן עובדות, כדי להבטיח שהודעת השגיאה מופיעה רק כשפעולה אינה

מודלקת, ולא באופן תמידי. עבור כל פעולה שנבדקת, מאפשרים את כל שאר הפעולות שלפניה במדפסת.

[הערה (לא שייכת לפתרון): יש כאן החלטה של אסטרטגיית בדיקות שצריכה להיות מוסברת ב-CTP; שהרי לכל פעולה יש 8 אפשרויות שונות של ההגדרות שצריכות כולן לגרום להופעת אותה שגיאה עבור הפעולה המוכיחה; מקרה הבדיקה שכאן בודק רק אחת משמונה אפשרויות אלה].

<b>הכנה (setup)</b>	
חבר את המדפסת למחשב	
הדלק את המדפסת	
בהגדרות, הדלק את כל הפעולות במדפסת;	
וודא שכל הפעולות אכן עובדות	
<b>צעדים</b>	
בעזרת ההגדרות, כבה את פעולת הפקס	
נסה לשלוח פקס	הודעה למסך: הפקס לא פעיל. לשליחת פקס, אפשר הדפסה, סריקה, שכפול ופקס "ב"הגדרות"
בעזרת ההגדרות, כבה את פעולת השכפול	
נסה לשכפל מסמך	הודעה למסך: שכפול מסמכים לא פעיל. לשכפול, אפשר לפחות הדפסה, סריקה, ושכפול "ב"הגדרות"
בעזרת ההגדרות, כבה את פעולת הסריקה	
נסה לשרוק מסמך	הודעה למסך: הסריקה לא פעילה. לסריקה, אפשר לפחות הדפסה וסריקה "ב"הגדרות"
בעזרת ההגדרות, כבה את פעולת ההדפסה	
נסה להדפיס מסמך	הודעה למסך: ההדפסה לא פעילה. להדפסה, אפשר לפחות הדפסה "ב"הגדרות"

(5)

א) לפי ההגדרות, הדפסה בצבע אפשרית רק עבור הדפסה וסריקה. כדי לאפשר שכפול בצבע, צריך מצד אחד להדליק שלוש פעולות (הדפסה, סריקה ושכפול) – אבל אז אי אפשר להדפיס בצבע כי שלוש פעולות מודלקות. גם: אם נדליק רק שכפול ופקס, אז על תיאורטית זה אומר שאפשר להדפיס בצבע. אבל כיוון ש"הדפסה" לא מודלקת בהגדרות, זה לא יתאפשר.

ב) **דרישה:** על המדפסת להדפיס מסמכים במהירות גבוהה

**סקלה:** דפים בשניה

**שיטת מדידה:**

- הנח במדפסת מספיק נייר להדפסת כל המסמך (לפחות 20 דפים)
- שלח את מסמך "print\_speed\_test.txt" להדפסה דרך חיבור ה-USB. חיבור: USB3.0
- מדוד את זמן ההדפסה מרגע שליחת המסמך ועד יציאת הדף המודפס האחרון. השתמש בשעון עצר
- חשב את זמן ההדפסה של כל דף

**מינימום:** פחות משניה לדף

**מטרה:** פחות מ 0.8 שניה לדף

**מעל המצופה:** פחות מ 0.6 שניה לדף

(6)

a. 6 בדיקות (הסבר: צריך 6 הרצות כדי לכסות את כל המודלים של הטלפונים. בכל שאר התנאים יש פחות ערכים, כך שנוכל לכסות אותם תוך הרצת ששת הבדיקות המתחייבות ממספר המודלים).

b. 16 בדיקות (הסבר: בדיקת קומבינציית ה-Base Choice תכסה ערך אחד מכל תנאי עבודה. לאחר מכן נשנה תנאי עבודה אחד, ונבדוק שוב; מספר הבדיקות לכיסוי הערכים שיש בתנאי עבודה אחד יהיה אם כך: מספר הערכים מינוס 1 – כי את אחד הערכים כבר כיסינו בבדיקה של ה-base choice. לאחר כיסוי תנאי עבודה אחד, נעבור לשני. גם שם נצטרך לבדוק מספר הערכים מינוס 1. וכו'. סך הבדיקות הוא אם כן: מספר כל הערכים בכל תנאי העבודה (20), פחות מספר תנאי העבודה (5) פלוס 1 (קומבינציית ה-Base Choice).

$$16 = 20 - 15 + 1$$

c. 720 בדיקות (הסבר: מכפלה של כל מספר הערכים בכל תנאי העבודה)

$$720 = 4 * 5 * 6 * 3 * 2$$

(7)

(א) **הנחה:** הפרמטר שישפיע במיוחד על המשחק שלנו הוא איכות המעבד הגרפי של הטלפון. אם המעבד חלש מידי, המשחק מוריד באופן אוטומטי את הרזולוציה של המסך על מנת לאפשר מהירות משחק סבירה.

לצורך הבדיקה נבחר שני מודלים: אחד עם המעבד הגרפי החלש ביותר והשני עם החזק ביותר. בשניהם נבחר את המסך הגדול ביותר. שיקולים: החלש ביותר יוודא שהמשחק אפשרי גם אם המעבד הגרפי לא הכי הכי, ושאכן הקביעה האוטומטית של הרזולוציה עובדת – גם אם המסך תומך ברזולוציה גבוהה יותר. החזק ביותר יוודא שהמשחק עובד היטב ברזולוציה הגבוהה ביותר.

[הערה: אפשר לבחור בכל מיני פרמטרים אחרים: מהירות וטכנולוגית רשת; OS; וכו' – העיקר שהסברתם בצורה משכנעת למה בחירה זו רלוונטית לבדיקת משחק הרשת שלכם]

(ב) לצורך הבדיקה נבחר את המודלים והפרמטרים שאנחנו מניחים שיהיו הדומיננטיים בשוק. זאת כיוון שבדיקות CI צריכות להיות מהירות (ב-CI מריצים רק חלק קטן מהבדיקות) ולכן נרצה לכוון לדברים שאם יפלו יהיו וודאי באג קריטי. הקריטיות של הבאגים מושפעת בין השאר מ"כמה משתמשים יפגעו מהבאג" ולכן הדגמים והפרמטרים הפופולריים הם המתאימים ביותר.

[שיקולים אחרים סבירים: מודלים שקל לאטמט; מודלים שמשתמשים ברשת ובטכנולוגיה הקיימת באיזור בו ממוקמת המעבדה, כך שאפשר להריץ בדיקות ממש על הרשת בלי לארגן סביבה מיוחדת; וכו' – העיקר שהסברתם וכו']

(8) מעבר לבדיקת אורך שם המשתמש אין כל בדיקה על התכולה שלו. לכן אפשר להעביר format string בתור שם המשתמש (למשל: ("%x%x%x"). מחרוזת זו "תופעל" כשהפקודה printf(username)

תתבצע. זה ידפיס למסך ערכים מתוך הזכרול של המחשב, שבעצם אסור לנו לגשת אליהם.

(9)  
(א)

```
def scanf_stub
    Return "a string longer than 20 characters"
```

```
def test_GetUsername
    ret = GetUsername()
    assert ret == ERROR
```

(ב)

```
#define FAIL 0
#define PASS 1
```

```
def scanf_stub
    Return "ShortUsername"
```

```
def RandomizeUserName_mock
    gTest_result = FAIL
    if username == "ShortUsername"
        gTest_result = PASS
```

```
def test_GetUsername
    username = ""
    ret = GetUsername()
    assert ret == SUCCESS
    assert gTest_result == PASS
```

(10)  
(א)

**Test 1:** Execute the "return -1" line  
myList = [1]

**Test 2:** Execute the "return 2\*a" line  
myList = [0, 0, 2, 10, 5]

**Test 3:** Execute the "return 2\*b" line  
myList = [0, 0, 2, 5, 5]

(ב) הפתרון כאן מתאר גם את מצב התנאים עבור כל מקרה בדיקה. במבחן היה צריך רק לתת את ערכי a ו-b בכל מקרה בדיקה.

a	b	a < b	a > b	a == b	a < 50	b > 100
5	10	T	T	F	T	F
-60	120	T	F	F	F	T
1	1	F	F	T	T	F

(ג)

הבודקת צודקת.

על מנת לקבל 100% כיסוי משפטים ל-foo() צריך להריץ אותה פעם אחת כאשר bar() מחזירה 0, ופעם כש bar() מחזירה 1.

כלומר, צריך לקרוא ל- bar() פעמיים: פעם אחת עם נתונים שיגרמו ל –

$(a == b) \mid\mid ((a < 50) \&\& (b > 100))$

לקבל ערך TRUE, ופעם אחת עם נתונים שיגרמו לפרדיקט זה לקבל ערך FALSE.

כלומר, פרדיקט זה יקבל 100% כיסוי החלטות.

גם אם שני הפרדיקטים האחרים ב-bar() יקבלו בשתי הקריאות ל-bar() את אותו ערך לוגי, עדיין נקבל כיסוי החלטות של 4/6 (66%) עבור bar().