אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

# התוכנית להנדסת תוכנה

מספר נבחן:_____

**הנחיות:**

1. יש לענות על השאלות רק בתוך שאלון הבחינה ורק בתוך המקומות המיועדים. תשובות מחוץ לשאלון או למקומות המיועדים — לא יבדקו.
2. אלא אם צוין אחרת במפורש, בכל שאלה יש לבחור תשובה אחת בלבד.
3. אם לפי דעתכם יש מספר תשובות נכונות, בחרו בתשובה הנכונה ביותר.

תאריך הבחינה: 21.02.2023
ל" שבט, תשפ"ג
שם המרצה: ד"ר אחיה אליסף
שם הקורס: הנדסת איכות תוכנה
מספר הקורס: 372-1-3501
סמסטר: א'
מועד: ב'
משך הבחינה: שעתיים
חומר עזר: מחשבון

## בהצלחה!

שאלה 1 (6 נקודות)

מנו והסבירו בקצרה שתי בעיות בבדיקות תוכנה באמצעות טסטים.
(אין הכוונה לסוג מסויים של בדיקות תוכנה אלה לעצם הרעיון של בדיקות תוכנה)

| |
|---|
| |
| |
| |
| |
| |
| |

שאלה 2 (12 נקודות) — בדיקות תוכנה

בבדיקות תוכנה יש ארבע רמות. מנו אותן לפי הסדר. מי מבצע אותן? מה עושים בכל שלב? למה לא מספיקים כל השלבים שלפני?

| |
|---|
| רמה 1 — בדיקות יחידה. |
| |
| רמה 2 |
| |
| רמה 3 |
| |
| רמה 4 |
| |

שאלה 3 (16 נקודות) — התאוריה של הלסטד

להלן מספר הגדרות מהתאוריה של הלסטד:

| | |
|---|---|
| נפח התוכנית : $V = N \log_2 \eta$ | מספר אופרטורים שונים בתוכנית : $\eta_1$ |
| $\check{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$ | מספר אופרנדים שונים בתוכנית : $\eta_2$ |
| סיבוכיות התוכנית : $D = (\eta_1/2) * (N_2/\eta_2)$ | מספר המופעים של אופרטורים בתוכנית : $N_1$ |
| ההשקעה הנדרשת בפיתוח התוכנית : $E = D \times V$ | מספר המופעים של אופרנדים בתוכנית : $N_2$ |
| הזמן הנדרש לפיתוח התוכנית : $T = E/\beta$. $\beta = 18$ | המילון של התוכנית : $\eta = \eta_1 + \eta_2$ |
| כמות הבאגים שנותרו במערכת : $B = V/3000$ | אורך התוכנית : $N = N_1 + N_2$ |

להלן מתודה:

```
1. public static int factorial(int n) {
2.   int result = 1;
3.   for (int i = 1; i <= n; i++) {
4.     result *= i;
5.   }
6.   return result;
7. }
```

סעיף א (10 נקודות)

חשבו את הפרמטרים השונים של הלסטד. לארבעה הראשונים יש לפרט איך הגעתם למספר זה

| |
|---|
| $\eta_1$= |
| $\eta_2$= |
| $N_1$= |
| $N_2$= |
| $\eta$ = |
| N= |
| V = |
| $\check{N}$ = |
| D = |
| E = |
| T = |
| B = |

סעיף ב (6 נקודות)

ציינו והסבירו בקצרה שתי חסרונות של שיטה זו

| |
|---|
| |
| |
| |
| |

שאלה 4 (30 נקודות) — Control-flow & Domain Testing

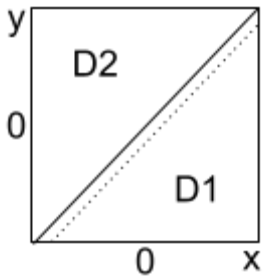
להלן מתודה:

```
1.  public void foo(int a, int b) {
2.    int z = a + b;
3.    if (a <= b + 5) {
4.      System.out.println("1");
5.      if (b > 2 * b) {
6.        System.out.println("2");
7.      } else if (b < 0) {
8.        System.out.println("3");
9.      } else {
10.       System.out.println("4");
11.     }
12.     if (z >= 2 * a) {
13.       System.out.println("5");
14.     }
15.     System.out.println("6");
16.   }
17.   System.out.println("7");
18. }
```

סעיף א (5 נקודות) — עליכם לצייר את ה control-flow graph עבור מתודה זו.

סעיף ב (10 נקודות)

שרטטו את ה domains של הפונקציה כפי שמופיע בדוגמה משמאל. יש למספר את הדומיינים השונים ולהוסיף קו מקווקו בצד הפתוח במקומות שהדומיין שפתוח.




דוגמה:
D1 פתוח

סעיף ג (5 נקודות) — כיתבו את התנאי של כל דומיין:

| |
|---|
| |
| |
| |
| |
| |
| |

סעיף ד (10 נקודות) — אתם חושדים שיש טעות בשורה 12 ושהתנאי צריך להיות $z + w >= 2 * a$ כאשר $w > 0$ (אתם לא יודעים את הערך המדוייק של w). הסבירו במילים ותנו דוגמה קונקרטית (עם קלטים) איך תאמתו את החשדות.

| |
|---|
| |
| |
| |
| |
| |
| |

שאלה 5 (21 נקודות) — Data flow testing

נתונה הפונקציה bar שמוצאת את הסכום המקסימלי של מספרים שליליים עוקבים במערך

```
1.  public int bar(int[] arr) {
2.    int sum = 0;
3.    int max = Integer.MIN_VALUE;
4.
5.    for (int i = 0; i < arr.length; i++) {
6.      if (arr[i] < 0) {
7.        sum += arr[i];
8.      } else {
9.        if (sum > max) {
10.         max = sum;
11.       }
12.       sum = 0;
13.     }
14.   }
15.   if (sum > max) {
16.     max = sum;
17.   }
18.
19.   return max;
20. }
```

סעיף א (10 נקודות) — עליכם לצייר את ה data-flow-graph עבור מתודה זו.

סעיף ב (11 נקודות)

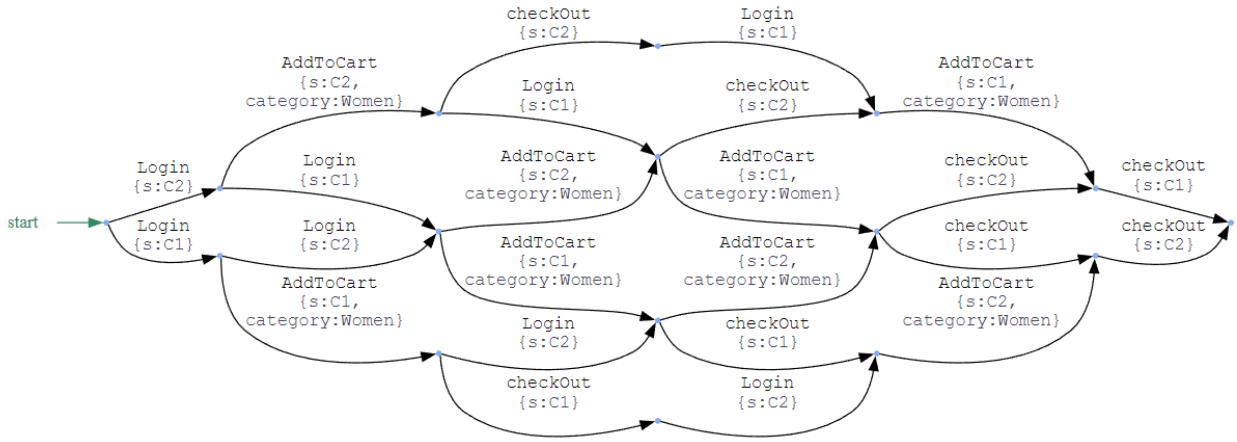
באילו קודקודים המשתנה max הוא global definition? _____.

כתבו סט מינימלי של מסלולים שלמים עבור המשתנה max והקריטריון All definition

יש לסמן בתוך המסלולים את ה definition-clear path שחייבים לעבור בהם.

שאלה 6 (15 נקודות) — model-based testing

בודקת תוכנה רצתה לבדוק תסריטים של הוספת מוצרים לעגלת קניות וסיום הקניה (checkout). היא כתבה מודל באמצעות שיטת model-based testing. הגרף הבא מתאר את מרחב המצבים האפשריים במודל שלה:



סעיף א (5 נקודות)

נאמר לכם שהמודל נכתב באמצעות fMBT או Provengo. האם ניתן לקבוע מהגרף באיזה כלי הבודקת השתמשה? נמקו.

| |
|---|
| |
| |

בשני הסעיפים הבאים אתם מתבקשים לבטל בגרף את האפשרות שאירוע

{AddToCart {s:C1, category:Women יקרה לפני אירוע {AddToCart {s:C2, category:Women

בשני הסעיפים ניתן להסביר במילים או לכתוב קוד.

סעיף ב (5 נקודות)

מה הייתם צריכים לעשות אם הייתם רוצים לבצע את זה במודל שנכתב באמצעות fMBT?

| |
|---|
| |
| |
| |
| |
| |

סעיף ג (5 נקודות)

מה הייתם צריכים לעשות אם הייתם רוצים לבצע את זה במודל שנכתב באמצעות Provengo?

| |
|---|
| |
| |
| |
| |
| |