

6

במרץ 2025

72 ט' מועד

## שאלון בחינת גמר

# 20441 - מבוא למדעי המחשב ושפת Java

## משך בחינה: 3 שעות

## בשאלון זה 21 עמודים

## מבנה הבחינה:

קראו בעיון את ההנחיות שלהלן:

\* עליכם לענות על חמש מתוך שש השאלות, בהתאם לסמסטר הלימוד שלכם.

\* כל התכניות צריכות להיות מתועדות היטב.

יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.

יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתכנית.

תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85 % מהניקוד.

\* יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.

תכנית לא יעילה לא תקבל את מלוא הנקודות.

\* אם ברצונכם להשתמש בתשובתכם בשיטה או במחלקה הכתובה בחוברת השקפים,

אין צורך שתעתיקו את השיטה או את המחלקה לקובץ הפתרונות הבחינה.

מספיק להפנות למקום הנכון,

ובלבד שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').

\* אין להשתמש במחלקות קיימות ב-Java, חוץ מאלו המפורטות בשאלות הבחינה.

\* יש לשמור על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרום מהציון.

\* בכתִיבת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה

אין להשתמש במשתנים גלובליים!

\* אפשר לתעד בעברית. אין צורך בתיעוד API.

\* על שאלות 3-5 יש לענות אך ורק בשאלון.

**חוברות השקפים 1-6, 7-12. אסור לכתוב כלום בתוך חוברות השקפים.**

**מותר לסמן עמודים בצבע או בדגלונים. אסור להכניס מחשב/מחשבון/**

**מכשיר אלקטרוני מכל סוג שהוא. אסורים עזרים וחומרים מקוונים.**

**אין להכניס חומר נוסף אחר מכל סוג. חומר העזר מודפס בלבד.**

## בהצלחה !!!

## חלק א – עליכם לענות על כל השאלות בחלק זה בקובץ התשובות

### שאלה 1 (25 נקודות)

כל מספר טבעי (שלם חיובי)  $num$  ניתן להציג כסכום של חזקות של מספר טבעי  $factor$  כלשהו, כאשר כל חזקה מופיעה פעם אחת לכל היותר.

#### דוגמאות:

- את המספר  $num = 9$  אפשר להציג כך:  
 $2^0 + 2^3 = 1 + 8 = 9$   $factor = 2$   
 $3^2 = 9$   $factor = 3$  וגם כך
- את המספר  $num = 19$  אפשר להציג כך:  
 $2^0 + 2^1 + 2^4 = 1 + 2 + 16 = 19$   $factor = 2$
- את המספר  $num = 28$  אפשר להציג כך:  
 $2^2 + 2^3 + 2^4 = 4 + 8 + 16 = 28$   $factor = 2$   
 $3^0 + 3^3 = 1 + 27 = 28$   $factor = 3$
- את המספר  $num = 42$  אפשר להציג כך:  
 $2^1 + 2^3 + 2^5 = 2 + 8 + 32 = 42$   $factor = 2$   
 $6^1 + 6^2 = 6 + 36 = 42$   $factor = 6$  וגם כך
- את המספר  $num = 73$  אפשר להציג כך:  
 $2^0 + 2^3 + 2^6 = 1 + 8 + 64 = 73$   $factor = 2$   
 $8^0 + 8^1 + 8^2 = 1 + 8 + 64 = 73$   $factor = 8$  וגם כך
- את המספר  $num = 273$  אפשר להציג כך:  
 $2^0 + 2^4 + 2^8 = 1 + 16 + 256 = 273$   $factor = 2$   
 $4^0 + 4^2 + 4^4 = 1 + 16 + 256 = 273$   $factor = 4$  וגם כך  
 $16^0 + 16^1 + 16^2 = 1 + 16 + 256 = 273$   $factor = 16$  וגם כך

#### שימו לב –

ברור שאפשר להציג כל מספר  $a$  כחזקה 1 של המספר  $a$  (כלומר  $a^1$ ), וגם כך:  
 $(a-1)^0 + (a-1)^1$ .

לדוגמא, את המספר 7 אפשר להציג כך:  $7 = 6^0 + 6^1 = 1 + 6$ . אנחנו לא מתכוונים לסכום הזה! את המספר 7 אנחנו רוצים להציג כך:  $7 = 2^0 + 2^1 + 2^2 = 1 + 2 + 4$

כתבו שיטה סטטית רקורסיבית המקבלת כפרמטר מספר שלם חיובי num כלשהו ומחזירה מהו המספר factor הגדול ביותר מבין המספרים 2 ל- num-2 כך שאפשר לקבל את num כסכום חזקות של factor, כאשר כל חזקה מופיעה לכל היותר פעם אחת. אם אין מספר כזה יוחזר 0. כמו כן, השיטה תדפיס את פירוט החזקות של factor שסכומן הוא num. ראו דוגמא בהמשך.

חתימת השיטה היא:

```
public static int maxFactor (int num)
```

בדוגמאות לעיל,

- אם  $num = 9$  – המספר שיוחזר יהיה 3 (כי הוא הגדול מבין 2, 3), ויודפס:  
9
- אם  $num = 73$  – המספר שיוחזר יהיה 8 (כי הוא הגדול מבין 2, 8), ויודפס:  
1      8      64
- אם  $num = 273$  – המספר שיוחזר יהיה 16 (כי הוא הגדול מבין 2, 4, 16), ויודפס:  
1      16      256
- אם  $num = 3$  – המספר שיוחזר יהיה 0 (כי אין factor אפשרי) ולא יודפס כלום.

כאן לא יוחזר הפקטור 2 ולא יודפס 2 בגלל שאנחנו לא מתייחסים לסכום של  $(n-1)^0 + (n-1)^1$  כפי שהסברנו לעיל.

הדרכה: (אין חובה להשתמש בה)

כדאי לכתוב שיטת עזר (רקורסיבית כמובן) שמקבלת את num ומספר נוסף factor, ובודקת אם אפשר להציג את num כסכום חזקות של factor.

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

- אם השיטה שתכתבו רק תחזיר את הגורם המקסימלי, אבל לא תדפיס את הצגת הצירופים, התשובה תקבל לכל היותר 17 נקודות.
- מותר להשתמש (אבל זה לא חובה!) בשיטות Math.max, Math.min, Math.sqrt מהמחלקה Math וכן בקבועים Integer.MIN\_VALUE, Integer.MAX\_VALUE מהמחלקה Integer.
- מותר להשתמש בהעמסת-יתר (Overloading)
- אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לתעד את מה שכתבתם!

## שאלה 2 (25 נקודות)

הגדרה: נאמר שהמספר השלם  $a$  מחלק את המספר השלם  $b$ , אם כשמחלקים את  $b$  ב-  $a$  אין שארית. כלומר,  $b \% a == 0$ . (לדוגמא, 3 מחלק את 12, 7 מחלק את 21 וכד').

כתבו שיטה סטטית יעילה שמקבלת כפרמטר מערך חד-ממדי  $arr$  המכיל מספרים שלמים חיוביים ממש בלבד, ומספר שלם חיובי  $num$ . השיטה צריכה לבדוק אם יש איבר במערך שהוא מחלק את הסכום של האיברים בתת-מערך שהוא נמצא באמצעו ומשמאלו ומימינו יש  $num$  איברים בכל צד. אם כן, השיטה צריכה להחזיר כמה איברים כאלו יש, אם אין, השיטה תחזיר 0.  
אם גודל המערך אינו מספיק כך שיהיו בו  $2 * num + 1$  איברים, השיטה תחזיר -1.

### דוגמאות:

נניח שהמערך  $arr$  הוא

0	1	2	3	4
25	10	5	20	40

- אם  $num = 1$ , אז יש 3 איברים פוטנציאליים שימלאו את התנאי.
  - האיבר  $arr[1]$ . סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש איבר אחד בכל צד הוא  $arr[0] + arr[1] + arr[2] = 25 + 10 + 5 = 40$  ואמנם, 10 מחלק את 40.
  - האיבר  $arr[2]$ . סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש איבר אחד בכל צד הוא  $arr[1] + arr[2] + arr[3] = 10 + 5 + 20 = 35$  ואמנם, 5 מחלק את 35.
  - האיבר  $arr[3]$ . סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש איבר אחד בכל צד הוא  $arr[2] + arr[3] + arr[4] = 5 + 20 + 40 = 65$  אבל, 20 אינו מחלק את 65.לכן השיטה תחזיר 2
- אם  $num = 2$ , אז יש איבר אחד פוטנציאלי למלא את התנאי.

האיבר  $arr[2]$ . סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש שני איברים בכל צד הוא  $arr[0] + arr[1] + arr[2] + arr[3] + arr[4] = 25 + 10 + 5 + 20 + 40 = 100$  ואמנם, 5 מחלק את 100.

לכן השיטה תחזיר 1

נניח שהמערך arr הוא

0	1	2	3	4	5	6	7	8	9	10
1	9	3	2	5	6	7	8	9	10	3

- אם  $num = 1$ , אז יש 5 איברים שימלאו את התנאי.

1.  $a[3]$  כי 2 מחלק את  $a[2]+a[3]+a[4] = 3+2+5 = 10$

2.  $a[5]$  כי 6 מחלק את  $a[4]+a[5]+a[6] = 5+6+7 = 18$

3.  $a[6]$  כי 7 מחלק את  $a[5]+a[6]+a[7] = 6+7+8 = 21$

4.  $a[7]$  כי 8 מחלק את  $a[6]+a[7]+a[8] = 7+8+9 = 24$

5.  $a[8]$  כי 9 מחלק את  $a[7]+a[8]+a[9] = 8+9+10 = 27$

- אם  $num = 2$ , אז יש 2 איברים שימלאו את התנאי.

1.  $a[6]$  כי 7 מחלק את  $a[4]+a[5]+a[6]+a[7]+a[8] = 5+6+7+8+9 = 35$

2.  $a[7]$  כי 8 מחלק את  $a[5]+a[6]+a[7]+a[8]+a[9] = 6+7+8+9+10 = 40$

- אם  $num = 3$ , אז יש 2 איברים שימלאו את התנאי.

1.  $a[4]$  כי 5 מחלק את 40

$a[1]+a[2]+a[3]+a[4]+a[5]+a[6]+a[7] = 9+3+2+5+6+7+8 = 40$

2.  $a[7]$  כי 8 מחלק את 48

$a[4]+a[5]+a[6]+a[7]+a[8]+a[9]+a[10] = 5+6+7+8+9+10+3 = 48$

**חתימת השיטה היא:**

```
public static int howManyDivisors (int [] arr, int num)
```

**שימו לב:**

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק, כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה, תקבל מעט נקודות בלבד.

ציינו מהי סיבוכיות זמן הריצה ומהי סיבוכיות המקום של השיטה שכתבתם. הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!

**חלק ב - את התשובות לשאלות 3, 4, 5 ו-6 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!**

**שאלה 3 (16 נקודות)**

נניח שהמחלקה Node שלהלן מממשת צומת בעץ בינרי.

```
public class Node
{
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int number)
    {
        _number = number;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }

    public void setNumber(int number) { _number = number; }
    public void setLeftSon(Node node) { _leftSon = node; }
    public void setRightSon(Node node) { _rightSon = node; }
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול **בעץ בינרי**.

בין השיטות במחלקה נתונות השיטות something ו-what הכתובות בעמודים הבאים:

```

public static int something(Node root)
{
    if (root == null)
        return Integer.MIN_VALUE;

    if (root.getLeftSon() == null && root.getRightSon() == null)
        return root.getNumber();

    int ans1 = something(root.getLeftSon());
    int ans2 = something(root.getRightSon());

    if (ans1 > ans2)
        return ans1 + root.getNumber();
    else
        return ans2 + root.getNumber();
}

public static boolean what(Node root, int num)
{
    if (num == 0 && root == null)
        return true;

    if (root == null)
        return false;

    boolean b1 = what(root.getLeftSon(), num - root.getNumber());
    boolean b2 = false;

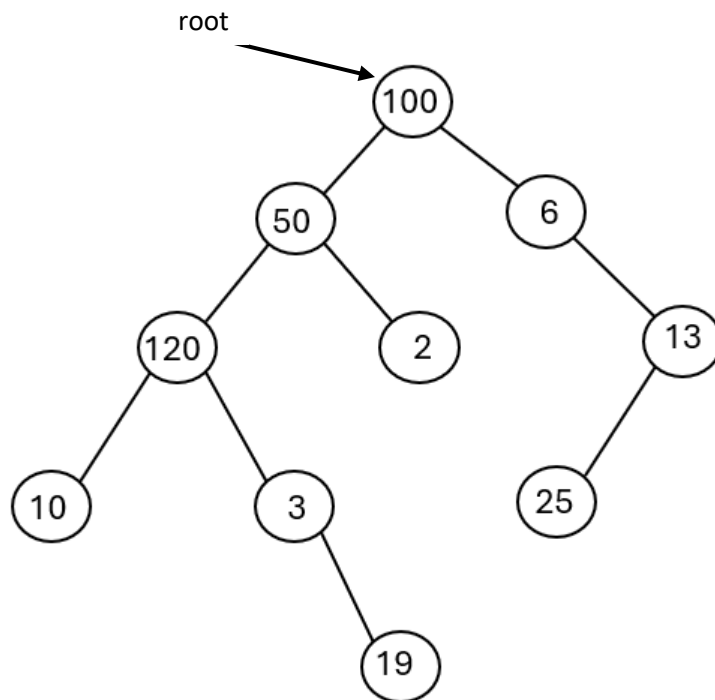
    if (!b1)
        b2 = what(root.getRightSon(), num - root.getNumber());

    if (b1 || b2)
        System.out.print(root.getNumber() + " ");

    return b1 || b2;
}

```

נתון העץ הבינרי הבא, ששורשו הוא root:



ענו על הסעיפים הבאים:

סעיף א (2 נקודות):

מה תחזיר השיטה `something` בעקבות הקריאה `BinaryTree.something(root)`?

התשובה היא:

---

סעיף ב (4 נקודות):

מה מבצעת השיטה `something` באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי `root`? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, לכתוב בקצרה מה משמעות הערך שהשיטה מחזירה. **התייחסו למקרי קצה.**

התשובה היא:

---

---

---

---



#### סעיף ג (4 נקודות):

מה תדפיס השיטה **what** בעקבות הפעלת שתי הפקודות להלן:

```
int num = BinaryTree.something(root);
```

```
BinaryTree.what(root, num);
```

התשובה היא:

---

#### סעיף ד (6 נקודות):

מה מבצעת השיטה **what** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי `root` ומספר שלם `num` כלשהו? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, לכתוב בקצרה השיטה מדפיסה. התייחסו למקרי קצה.

התשובה היא:

---

---

---

---

---

---

---

## שאלה 4 (16 נקודות)

נתונה המחלקה `IntNode` הבאה, המייצגת איבר ברשימה מקושרת חד-סטרית המכילה מספרים שלמים:

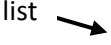
```
public class IntNode
{
    private int _value;
    private IntNode _next;

    public IntNode(int val, IntNode n) {
        _value = val;
        _next = n;
    }

    public int getValue()           { return _value; }
    public IntNode getNext()        { return _next; }
    public void setValue(int v)     { _value = v;    }
    public void setNext(IntNode node) { _next = node; }
}
```

נתונה רשימה מקושרת חד-סטרית, הממומשת בעזרת המחלקה `IntList` שמכילה את השיטה **what** הכתובה להלן. אפשר להניח ששני הפרמטרים `n` ו-`m` שהשיטה **what** מקבלת, הם בתחום בין 1 לאורך הרשימה. אפשר להניח שיש שיטות נוספות במחלקה.

בשאלות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים. כאשר המספר השמאלי מסמן את ראש הרשימה.

למשל הרשימה `list`: `1 → 2 → 3 → null`  תסומן כך: `{1, 2, 3}`

```
public class IntList
{
    private IntNode _head;

    public IntList()           { _head = null; }
    public IntList(IntNode node) { _head = node; }

    public String toString()   { ... }

    // הניחו שהשיטה ממומשת ומחזירה מחרוזת תווים המייצגת את הרשימה.
    // {1, 2, 3} נראית כך:

```

- המשך השאלה בעמוד הבא -

```

public IntList what (int m, int n)
{
    IntNode head = _head;

    if (m > n)
        return new IntList(head);

    IntNode p1 = null;
    IntNode p2 = head;

    for (int i = 1; p2 != null && i < m; i++)
    {
        p1 = p2;
        p2 = p2.getNext();
    }

    IntNode a1 = p2;
    IntNode a2 = null;

    for (int i = 1; p2 != null && i <= n - m + 1; i++)
    {
        IntNode ptr = p2.getNext();
        p2.setNext(a2);
        a2 = p2;
        p2 = ptr;
    }

    if (a1 != null)
    {
        a1.setNext(p2);
        if (p1 != null)
            p1.setNext(a2);
        else
            head = a2;
    }
    return new IntList(head);
}

//end of class IntList

```

אפשר להניח ששני הפרמטרים  $m$  ו-  $n$  שהשיטה `what` מקבלת, הם בתחום בין 1 לאורך הרשימה.

### סעיף א (2 נקודות)

נפעיל את השיטה **what** על הרשימה `list` הבאה:

`{81, 17, 8, 60, 101, 100, 24, 8, 9, 10, 11, 12}`, כאשר הפרמטרים `m = 4` ו-`n = 9`,

מה יודפס לאחר ביצוע הפקודה

```
System.out.println (list.what(4, 9));
```

התשובה היא:

---

### סעיף ב (2 נקודות)

נתונה הרשימה `list1` הבאה: `{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`

נניח שנתונה לנו השיטה `length()` המחזירה את אורך הרשימה עליה היא מופעלת.

כמו כן, נתון קטע הקוד הבא:

```
int len = list1.length();
for (int i=1; i<=len/2; i++)
{
    list1 = list1.what(i, len-i+1);
}
System.out.println(list1);
```

מה יודפס לאחר ביצוע קטע הקוד לעיל?

התשובה היא:

---

### סעיף ג (2 נקודות)

שנו את קטע הקוד מסעיף ב כך שהרשימה שתודפס תהיה הפוכה מהרשימה שהודפסה כתשובה לסעיף ב.

מהם השינויים המינימליים שצריך לעשות לשם כך?

כתבו את תשובתכם בעמוד הבא בשאלון (ורק שם!).

לא בהכרח כל השורות הריקות יתמלאו!

שימו לב שכדאי מאד להיצמד לשיטה `find` ולשנות אותה, ולא לכתוב שיטה מחדש.

**התשובה היא:**

```
int len = list1.length();
```

---

---

---

---

---

---

---

---

---

```
System.out.println(list1);
```

**סעיף ד (3 נקודות)**

האם יש רשימה list2 בת 7 איברים לפחות, **כאשר לפחות 4 מתוכם שונים זה מזה**, כך שאם נריץ את קטע הקוד מסעיף ב, הדפסת הרשימה לפני ואחרי קטע הקוד תהיה זהה?  
אם יש כזו, כתבו דוגמא לרשימה כזו.

אם לא, נמקו מדוע.

**התשובה היא:**

---

**סעיף ה (3 נקודות)**

האם יש רשימה list2 בת 7 איברים לפחות, **כאשר לפחות 4 מתוכם שונים זה מזה**, כך שאם נריץ את קטע הקוד מסעיף ב, **כל** האיברים ברשימה ישנו את מקומם?  
אם יש כזו, כתבו דוגמא לרשימה כזו.

אם לא, נמקו מדוע.

התשובה היא:

---

---

---

---

#### סעיף ו (4 נקודות)

מה מבצעת השיטה **what** באופן כללי, כאשר היא פועלת על רשימה list כלשהי, ומקבלת כפרמטרים מספרים m ו-n שלמים כלשהם?  
שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת.  
התייחסו למקרי קצה.

התשובה היא:

---

---

---

---

---

---

לפניכם שתי שאלות. יש לענות על שאלה אחת מבין השתיים בהתאם לסמסטר הלימוד שלך.

- אם למדת את הקורס בסמסטר 2024 או 2024 ב – חובה עליך לענות על שאלה 5
- אם למדתם את הקורס בסמסטר 2025 או לפני 2024 – חובה עליך לענות על שאלה 6

**לתשומת לבך! באחריות הסטודנט/ית להקפיד על בחירת השאלה המתאימה. שאלה לא מתאימה לא תיבדק!**

**שאלה 5 (18 נקודות) – רק למי שלמד/ה את הקורס בסמסטר 2024/א/ב**

במחלקה Q5 נתונה השיטה הסטטית secret הבאה, המקבלת כפרמטרים מערך חד-ממדי a המלא במספרים שלמים 1 ו-0 בלבד, ומספר שלם אי-שלילי k.

```
public class Q5
{
    public static int secret(int []a, int k)
    {
        int res = 0, i = 0, j = 0, temp = 0;
        while (i < a.length)
        {
            if (a[i] == 0)
            {
                temp++;
                if (temp > k)
                {
                    while (a[j] != 0)
                        j++;
                    j++;
                    temp--;
                }
            }
            if (i-j+1 > res)
                res = i-j+1;
            i++;
        }
        return res;
    }
} //end of class Q5
```

נתון המערך  $a$  הבא:

0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1	1	0	1	1	1	1	0	0

ענו על הסעיפים הבאים:

סעיף א (3 נקודות):

מה תחזיר השיטה  $\text{secret}$  בעקבות הקריאה  $\text{secret}(a, 0)$ ? נמקו את תשובתכם.

התשובה היא:

---

---

---

סעיף ב (3 נקודות):

עבור איזה תחום ערכים של  $k$ , הקריאה  $\text{secret}(a, k)$  תחזיר את הערך הכי גדול (מתוך כל הערכים שיוחזרו מקריאה לשיטה עם כל ערכי ה- $k$  האפשריים)? נמקו את תשובתכם.

התשובה היא:

---

---

---

סעיף ג (3 נקודות):

מלאו את המערך להלן בערכים כך שלכל  $k$  בין 0 ל-9, הערך שיוחזר מהקריאה לשיטה  $\text{secret}(a, k)$  יהיה שווה ל- $k$ .  
אם זה לא יתכן, נמקו מדוע.

התשובה היא:

0	1	2	3	4	5	6	7	8	9

---

---



### סעיף ד (3 נקודות):

מלאו את המערך להלן בערכים כך שלכל  $k$  בין 0 ל-9, הערך שיוחזר מהקריאה לשיטה  $\text{secret}(a, k)$  יהיה זהה.  
אם זה לא יתכן, נמקו מדוע.

### התשובה היא:

0	1	2	3	4	5	6	7	8	9

### סעיף ה (6 נקודות)

מה מבצעת השיטה **secret** באופן כללי, כאשר היא מקבלת כפרמטרים מערך חד-ממדי  $a$  מלא באחדים (1) ובאפסים (0), ומספר שלם  $k$ .  
שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, כתבו בקצרה מה משמעות הערך שהשיטה מחזירה כשהיא מקבלת כפרמטרים מערך ומספר שלם כפי שהוגדרו לעיל. **התייחסו למקרי קצה.**

### התשובה היא:

---

---

---

---

---

---

**שאלה 6 (18 נקודות) - רק למי שלמד/ה את הקורס בסמסטר 2025 או לפני 2024**

נתון פרויקט שהוגדרו בו המחלקות האלו:  $E, F, K, L, N, P$

בפרויקט מומשה השיטה  $f()$  בשלוש מחלקות והשיטה  $g()$  במחלקה אחת בלבד.

השיטה `equals` לא מומשה באף אחת מהמחלקות.

להלן ינתנו פרטים שונים שיעזרו לכם להבין מהם הקשרים בין המחלקות, ובאילו מחלקות מומשו השיטות  $f()$  ו  $g()$ . (הערך המוחזר של שתי השיטות הוא מטיפוס `String`)

במחלקה נוספת באותו פרויקט, בשם `Driver` נכתב קטע הקוד הבא בשיטה `main` אשר יוצר את העצמים הבאים, והוא תקין.

```
K[] arr = new K[5];
arr[0] = new N();
arr[1] = new P();
arr[2] = new L();
arr[3] = new F();
N n1 = new L();
E e = new E();
L lf = new F();
```

כמו כן, באותה מחלקה `Driver` בהמשך השיטה `main` נכתבו פקודות נוספות (כל פקודה לא תלויה בפקודות האחרות), בהערה בסוף כל שורה, כתוב מה קרה בעקבות הוספת פקודה זו:

1.	<code>P pn = new N();</code>	// compilation error
2.	<code>E ef = new F();</code>	// compilation error
3.	<code>F fe = new E();</code>	// compilation error
4.	<code>N np = new P();</code>	// compilation error
5.	<code>arr[2].f();</code>	// compilation error
6.	<code>n1.f();</code>	// compilation error
7.	<code>System.out.println (((L)arr[2]).f());</code>	// prints f3
8.	<code>System.out.println (((P)arr[1]).f());</code>	// prints f1
9.	<code>System.out.println (((E)arr[2]).f());</code>	// run-time error
10.	<code>System.out.println (e.f());</code>	// prints f3
11.	<code>System.out.println (((F)lf).f());</code>	// prints f2
12.	<code>System.out.println (arr[2].g());</code>	// prints g1

**סעיף א (6 נקודות)**

**ציירו עץ ירושה אפשרי של כל המחלקות:**



לדוגמא, אם המחלקה B יורשת ישירות מהמחלקה A, ציירו זאת כך:  
ציירו כאן:

**סעיף ב (6 נקודות)**

כתבו באילו מחלקות מופיעה השיטה  $f()$ .

בכל מחלקה כתבו את המימוש של השיטה  $f()$ .

**התשובה היא:**

---

---

---

---

---

---

---

---

## סעיף ג (6 נקודות)

בהמשך השיטה main במחלקה Driver נכתבו הפקודות הבאות, והן **תקינות**.

```
E e2 = new E();
```

```
Object o1 = new Object();
```

```
Object o2 = new E(o1);
```

**לכל אחת מהפקודות הבאות**, סמנו בצורה ברורה ומפורשת אם היא תקינה או גורמת לשגיאת קומפילציה או גורמת לשגיאת ריצה

אם הפקודה תקינה – כתבו מה יודפס (אם יש מה שיודפס).

אם הפקודה גורמת לשגיאת קומפילציה או שגיאת זמן ריצה – כתבו מהי הבעיה.

**הפקודות אינן תלויות זו בזו.**

a) `arr[3] = 1f;`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה

הסבר: \_\_\_\_\_

\_\_\_\_\_

b) `1f = arr[3];`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה

הסבר: \_\_\_\_\_

\_\_\_\_\_

c) `System.out.println(o2 instanceof Object);`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה

הסבר: \_\_\_\_\_

\_\_\_\_\_

d) `System.out.println(o2 instanceof E);`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה

הסבר: \_\_\_\_\_

\_\_\_\_\_

e) `System.out.println(e2.equals(o2));`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה

הסבר: \_\_\_\_\_

\_\_\_\_\_

f) `System.out.println(((E)o1).equals(e2));`

תקין / שגיאת קומפילציה / שגיאת זמן ריצה

הסבר: \_\_\_\_\_

\_\_\_\_\_

**ב ה צ ל ח !**