



דוח משוב לסטודנט

מזהה סטודנט N102415087
 תאריך בבחינה ים חמישי, 6 במרץ 2025
 מזהה קורס 20441
 שם קורס מבוא למדעי המחשב ושפת Java
 תמר ולניר מרצה

ציון מבחן טופי

ציון מבחן מקורי

ניקוד שאלות פתוחות

100.00**100.00****100.00**

סיכום

מספר שאלה	ניקוד	ניקוד מירבי	ניקוד מירבי
1	25.00	25.00	25.00
2	25.00	25.00	25.00
3	16.00	16.00	16.00
4	16.00	16.00	16.00
5	18.00	18.00	
6	18.00	18.00	
7	10.00		

מוס' שאלון - 463

6

במרץ 2025

ו' באדר תשפ"ה

92 מוס' מועד

סמסטר 2025 א

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפה Java

משך בחינה: 3 שעות

בשאלון זה 21 עמודים

מבנה הבחינה:

- קראו בעין את הנקודות שללן:
- * עליכם לענות על חמישה מטוער שיש השאלות, בהתאם לסמסטר הלימוד שלכם.
 - * כל התכניות צריכות להיות מ透ודות היטב.
 - יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.
 - יש לבחור בשמות מושמעותיים למשתנים, לפונקציות ולקבועים שבתכנית.
 - תכנית שלא ת透וד כנדרש לעיל תקבל לכל היתר % 85 מהניקוד.
 - * יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.
 - תכנית לא ייעלה לא לקבל את מלאה הנקודות.
 - * אם ברצונכם להשתמש בתשובותכם בשיטה או בחלוקת הכתובה בחוברת השקפים, אין צורך שתעתיקו את השיטה או את החלוקת מהבחינה.
 - מספיק להפנות למקומות הנכוון,
 - ובלבד שההפנייה תהיה מדויקת (פרמטרים, מיקום וכו').
 - * אין להשתמש בחלוקת קיימות ב- Java , חוץ מאלו המפורטוות בשאלות הבחינה.
 - * יש לשמר על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרוע מהzion.
 - * בכתיבת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה
 - אין להשתמש במשתנים גלובליים!
 - * אפשר לטעyd בעברית. אין צורך בתיעוד API .
 - * על שאלות 3-5 יש לענות אך ורק בשאלון ולא בחוברת הבחינה!

חומר עזר:

חוורות השקפים 6-1, 7-12. אסור לכתוב כלום בתוך חוותות השקפים.

מותר לסמן עמודים בכתב או בדגלונים.

אין להכניס מחשב או מכשיר אלקטרוני מכל סוג שהוא.

.

בהצלחה !!!

**הצדiron
למשגיח את השאלון
וכל עזר אחר שקיבלתם בתוך מחברת התשובות**



1970-02-07

1970-02-07

חלק א – עלייכם לענות על כל השאלות בחלק זה במחברת

הבחינה

שאלה 1 (25 נקודות)

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

דוגמאות:

- את המספר $9 = 1 + 8 = 9$ ניתן לאפשר להציג כך:

$$2^0 + 2^3 = 1 + 8 = 9$$

factor = 2

$$3^2 = 9$$

וגם כך factor = 3

- את המספר $19 = 1 + 2 + 16 = 19$ ניתן לאפשר להציג כך:

$$2^0 + 2^1 + 2^4 = 1 + 2 + 16 = 19$$

factor = 2

- את המספר $28 = 1 + 27 = 28$ ניתן לאפשר להציג כך:

$$2^2 + 2^3 + 2^4 = 4 + 8 + 16 = 28$$

factor = 2

$$3^0 + 3^3 = 1 + 27 = 28$$

וגם כך factor = 3

- את המספר $42 = 1 + 8 + 32 = 42$ ניתן לאפשר להציג כך:

$$2^1 + 2^3 + 2^5 = 2 + 8 + 32 = 42$$

factor = 2

$$6^1 + 6^2 = 6 + 36 = 42$$

וגם כך factor = 6

- את המספר $73 = 1 + 8 + 64 = 73$ ניתן לאפשר להציג וכך:

$$2^0 + 2^3 + 2^6 = 1 + 8 + 64 = 73$$

factor = 2

$$8^0 + 8^1 + 8^2 = 1 + 8 + 64 = 73$$

וגם כך factor = 8

- את המספר $273 = 1 + 16 + 256 = 273$ ניתן לאפשר להציג וכך:

$$2^0 + 2^4 + 2^8 = 1 + 16 + 256 = 273$$

factor = 2

$$4^0 + 4^2 + 4^4 = 1 + 16 + 256 = 273$$

וגם כך factor = 4

$$16^0 + 16^1 + 16^2 = 1 + 16 + 256 = 273$$

וגם כך factor = 16

שימוש לב –

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

לדוגמא, את המספר 7 אפשר להציג כך: $6^0 + 6^1 + 6^0 = 1 + 6 + 1 = 7$. אנחנו לא מתכוונים לסכום זה! 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** הגדל ביותר מבין המספרים 2 ל-`num` 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** שכוכם הוא `num`. ראו דוגמא בהמשך.

חתימת השיטה היא:

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

בדוגמאות לעיל,

- אם $9 = num$ – המספר שיוחזר יהיה 3 (כי הוא הגדל מבין 3, 2), וידפס:
9
- אם $73 = num$ – המספר שיוחזר יהיה 8 (כי הוא הגדל מבין 8, 2), וידפס:
1 8 64
- אם $273 = num$ – המספר שיוחזר יהיה 16 (כי הוא הגדל מבין 16, 4, 2) וידפס:
1 16 256
- אם $3 = num$ – המספר שיוחזר יהיה 0 (כי אין **factor** אפשרי) ולא יודפס כלום.
- כאן לא יוחזר הפקטור 2 ולא יודפס 2 1 בגלל שאנחנו לא מתיחסים לסכום של $1^{(1-h)} + 0^{(1-h)}$ כפי שהסבירנו לעיל.

הדרך: (אין חובה להשתמש בה)

כדי לכתוב שיטת עדר (רקורסיבית כמובן) שמקבלת את `num` ומספר נוסף **factor**, ובודקת אם אפשר להציג את `num` כסכום חזקות של **factor**.

השיטה צריכה להיות רקורסיבית ללא שימוש בולואות כלל. כך גם כל שיטות העדר שתכתבו (אם תכתבו) לא יכולות להכיל לולואות.

- אם השיטה שתכתבו רק תחזיר את הגורם המקורי, אבל לא תדפיס את הצגת היצירופים, התשובה תתקבל לכל היותר 17 נקודות.
- מותר להשתמש (אבל זה לא חובה!) בשיטות `Math.sqrt` , `Math.min` , `Math.max` מהמחלקה `Math` וכן קבועים `Integer.MIN_VALUE` ,`Integer.MAX_VALUE` מהמחלקה `Integer`.
- מותר להשתמש בעומסת-יתר (**Overloading**)
- אין צורך לדאוג ליעילות השיטה! אבל כמובן צריך לשימוש לב לא לעשות קרייאות רקורסיביות מיותרות!
- אל תשכחו לטעyd את מה שתכתבتم!

שאלה 2 (25 נקודות)

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

כתבו שיטה סטטית **יעילה** שמקבלת כפרמטר מערך חד-ממדי **zzm** המכיל מספרים שלמים חיוביים ממש בלבד, ומספר שלם חיובי **wut**. השיטה צריכה לבדוק אם יש איבר במערך שהוא מחלק את הסכום של האיברים בתת-מערך שהוא נמצא באמצעותו ומשמאלו ומימינו יש **wut** איברים בכל צד. אם כן, השיטה צריכה להחזיר כמה איברים całואיש, אם אין, השיטה תחזיר 0.

אם גודל המערך אינו מספיק כך שייהי בו $1 + \text{num}^2$ איברים, השיטה תחזיר -1.

דוגמאות:

נניח שהמערך **zzm** הוא

0	1	2	3	4
25	10	5	20	40

- אם $1 = wut$, אז יש 3 איברים פוטנציאליים שימלאו את התנאי.
1. האיבר $[1]_{zzm}$. סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש איבר אחד בכל צד הוא $25 + 10 + 5 = 40 = arr[0] + arr[1] + arr[2]$.
ובאמת, 10 מחלק את 40.
- 2. האיבר $[2]_{zzm}$. סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש איבר אחד בכל צד הוא $10 + 5 + 20 = 35 = arr[1] + arr[2] + arr[3]$.
ובאמת, 5 מחלק את 35.
- 3. האיבר $[3]_{zzm}$. סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש איבר אחד בכל צד הוא $5 + 20 + 40 = 65 = arr[2] + arr[3] + arr[4]$.
אבל, 20 אינו מחלק את 65.

לכן השיטה תחזיר 2

- אם $2 = wut$, אז יש איבר אחד פוטנציאלי למלא את התנאי.
האיבר $[2]_{arr}$. סכום התת-מערך שהוא באמצעו ומשמאלו ומימינו יש שני איברים בכל צד הוא $100 = 100 + 40 + 25 + 10 + 5 = arr[0] + arr[1] + arr[2] + arr[3] + arr[4]$.
ובאמת, 5 מחלק את 100.

לכן השיטה תחזיר 1

גַּתְּהָרָה כִּינְתָּן

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

$$\begin{array}{ll}
 \text{כי } 2 \text{ מחלק את } 10 & a[3] .1 \\
 a[2]+a[3]+a[4] = 3+2+5 = 10 & \\
 \text{כי } 6 \text{ מחלק את } 18 & a[5] .2 \\
 a[4]+a[5]+a[6] = 5+6+7 = 18 & \\
 \text{כי } 7 \text{ מחלק את } 21 & a[6] .3 \\
 a[5]+a[6]+a[7] = 6+7+8 = 21 & \\
 \text{כי } 8 \text{ מחלק את } 24 & a[7] .4 \\
 a[6]+a[7]+a[8] = 7+8+9 = 24 & \\
 \text{כי } 9 \text{ מחלק את } 27 & a[8] .5 \\
 a[7]+a[8]+a[9] = 8+9+10 = 27 &
 \end{array}$$

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

$$\begin{array}{ll}
 \text{כי } 7 \text{ מחלק את } 35 & a[6] .1 \\
 a[4]+a[5]+a[6]+a[7]+a[8] = 5+6+7+8+9 = 35 & \\
 \text{כי } 8 \text{ מחלק את } 40 & a[7] .2 \\
 a[5]+a[6]+a[7]+a[8]+a[9] = 6+7+8+9+10 = 40 &
 \end{array}$$

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

$$\begin{array}{ll}
 \text{כי } 5 \text{ מחלק את } 40 & a[4] .1 \\
 a[1]+a[2]+a[3]+a[4]+a[5]+a[6]+a[7] = 9+3+2+5+6+7+8 = 40 & \\
 \text{כי } 8 \text{ מחלק את } 48 & a[7] .2 \\
 a[4]+a[5]+a[6]+a[7]+a[8]+a[9]+a[10] = 5+6+7+8+9+10+3 = 48 &
 \end{array}$$

חתימת השיטה היא:

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

שימוש לב:

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

ציין מהי סיבוכיות זמן הריצה ומהי סיבוכיות המקום של השיטה שתכתבם.

הסבירו תשובתכם. אל תשחחו לטעד את מה שתכתבם!

חלק ב - את התשובות לשאלות 3, 4, 5 ו-6 יש לכתוב על גבי השאלה. לא נבדוק תשובות שייכתו למקום אחר!

שאלה 3 (16 נקודות)

נניח שהמחלקה Node שלהן מימוש 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;

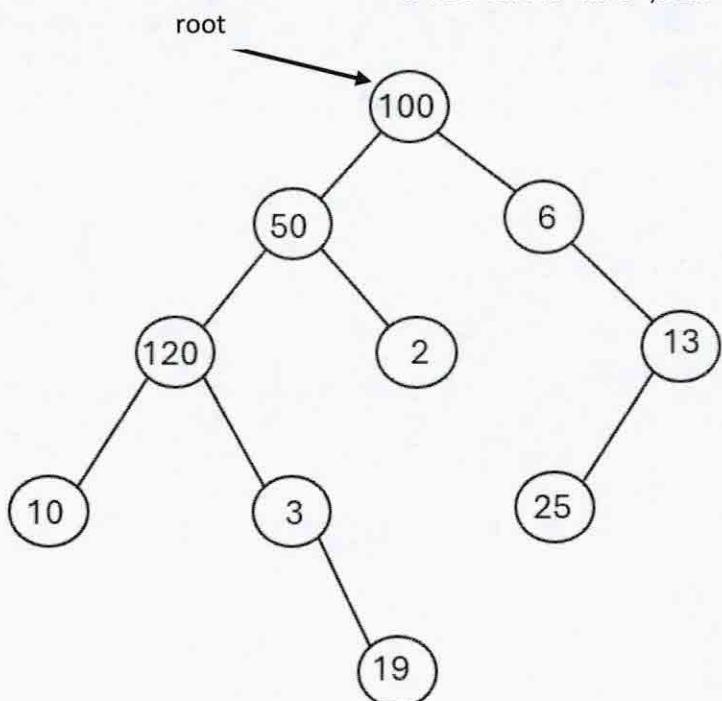
    if (!b1)
        b2 = what(root.getRightSon(), num-root.getNumber());

    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 (root)?

התשובה היא:

$$19 = 3 < 25 < 100 = 120 < 29$$



סעיף ב (4 נקודות):

מה מבצעת השיטה something באופן כללי, כשהיא מקבלת כפרמטר שורש של עץ ביני root? שימו לב, עליו לחתה 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. התויחסו למקרי קצה.

התשובה היא:

ארחיזג צויר הבדיקה נסובב גיאזיגון סטטוס גיאזיגון סטטוס
או. נזיגון נזיגון נזיגון נזיגון נזיגון
כיאזיגון נזיגון נזיגון נזיגון נזיגון
MIN_VALUE



סעיף ג (4 נקודות):

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

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

התשובה היא:

19, 3, 120, 50, 100



סעיף ד (6 נקודות):

מה מבצעת השיטה **what** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בineri **root** ומספר שלם **num** כתשובה? שימו לב, עליכם לתת תיאור ממצה של מה עשו השיטה באופן כללי, ולא תיאור של מה עשו כל שורה בשיטה, או איך היא מבצעת זאת. כמובן, לכתוב בקצרה השיטה מדפסה. התיחסו למקרים קצה.

התשובה היא:

השיטה **what** מקבלת שני פרמטרים: שורש העץ **root** ומספר **num**.
השיטה מודפסת את כל הnosrootnodes של העץ **root**, כאשר כלノוד נסמן
במספר **num**.



16
(3)

שאלה 4 (16 נקודות)

נתונה המחלקה `IntNode` הבאה, המציגת איבר בראשימה 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, המוממשת בעזרת המחלקה `IntList` שמכילה את השיטה **what** הכתובה להלן. אפשר להניח שני הפרמטרים `ch` ו- `w` שהשיטה **what** מקבלת, הם בתחום בין 1 לארך הרשימה. אפשר להניח שיש שיטות נוספות במחלקה.

בשאלות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגרים מסולסים. כאשר המספר **השמאלי** מסמן את ראש הרשימה.

למשל הרשימה `list`: `null → 1 → 2 → 3 → 1` תסמן 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 מחזוזת תווים המציגת את הרשימה. //
    המחרוזת נראית כך: {1, 2, 3} //
```

- 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 שני הפרמטרים m ו- n שהשיטה **what** מקבלת, הם בתחום בין 1 ל-אורך הרשימה.

סעיף א (2 נקודות)

נפעיל את השיטה **what** על הרשימה **list** הקיימת:
 $\{81, 17, 8, 9, 8, 24, 100, 101, 60, 10, 11, 12\}$, כאשר הפרמטרים $4 = m$ ו- $n = 9$
 מה יודפס לאחר ביצוע הפקודה

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

התשובה היא:

(81, 17, 8, 9, 8, 24, 100, 101, 60, 10, 11, 12)



12

סעיף ב (2 נקודות)

נתונה הרשימה **list1** הקיימת: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
 נניח שננתונה לנו השיטה **()length** המחזיר את אורך הרשימה עלייה 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:

{10, 2, 8, 4, 6, 5, 7, 3, 9, 13}



סעיף ג (2 נקודות)

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

מהם השינויים המינימליים שצרכי לעשות לשם כך?

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

לא בהכרח כל השורות הריקות יתملאו!

איך זה יכול?

שימוש לב שבדאי מאד להיזמד לשיטה **push** ולשנות אותה, ולא לכתוב שיטה חדש.

התשובה היא:

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



סעיף ד (3 נקודות)

אם 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 צזו, כתבו דוגמא לרשימה צזו.
אם לא, נמקו מדוע.

התשובה היא:

72455431 ~~123456789~~ 34



סעיף ה (3 נקודות)

אם 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 צזו, כתבו דוגמא לרשימה צזו.
אם לא, נמקו מדוע.

התשובה היא:

*Now we have great power, N
just as I said. I thank you very much. 15*



סעיף I (4 נקודות)

מה מבצעת השיטה **what** באופן כללי, כאשר היא פועלת על רשימה `list` כלשהי, ומתקבלת כפרמטרים מספריים `m` ו- `n` שלמים כלשהם?

עושה כל שורה בשיטה, או איך היא מבצעת זאת. שמו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה

התיחסו למקרי קצה.

התשובה היא:

מג' פון בֶּן־בָּרוּךְ

\rightarrow C_7J $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ \rightarrow $\text{N}(\text{C}_7\text{J})$ $m=2, n=4$ \rightarrow C_7J $n = 3$

$\sqrt{b} > 0$, $15 m > b$ $\Rightarrow c \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$



16
(4)

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

- אם למדת את הקורס בסמינר 2024 א או 2024 ב – חובה עלייך לענות על שאלה 5
- אם למדתם את הקורס בסמינר 2025 א או לפני 2024 א – חובה עלייך לענות על שאלה 6

תשומת לב! באחריות הסטודנט/ית להקפיד על בחירת השאלה המתאימה. שאלה לא מתאימה לא תיבדק!

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

במחלקה Q5 נתונה השיטה הステיטית `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 `secret` בעקבות הקריאה $(0, \text{secret}(a))$? נמקו את תשובהיכם.

התשובה היא:

סעיף ב (3 נקודות):

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

התשובה היא:

סעיף ג (3 נקודות):

מלאו את המערך להלן בערכים כך שלכל k בין 0 ל- 9 , הערך שיוחזר מקריאה לשיטה `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 קר שלכל k בין 0 ל- 9, הערך שיוחזר מהקriteria לשיטה $\text{secret}(a, k)$ יהיה זהה.
אם זה לא נכון, 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 .
שים לב, עליכם לתת תיאור מזכה של מה עשוה השיטה באופן כללי, ולא תיאור של מה עשוה 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. **התיכון למקרי קצה.**

התשובה היא:

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

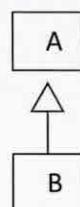
נתון פרויקט שהוגדרו בו המחלקות הלו: E, F, K, L, N. בפרויקט מומשא השיטה f() בשלוש מחלקות והשיטה g() במחלקת אחת בלבד. השיטה equals לא מומשא באף אחת מהמחלקות. להלן נתונים פרטיים שונים שיעזרו לכם להבין הקשרים בין המחלקות, ובאיו מחלקות מומשו השיטות f() ו g(). (הערך המוחזר של שתי השיטות הוא מטיפוס String). במחלקה נוספת 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 nl = new L();
E e = new E();
L lf = new F();
```

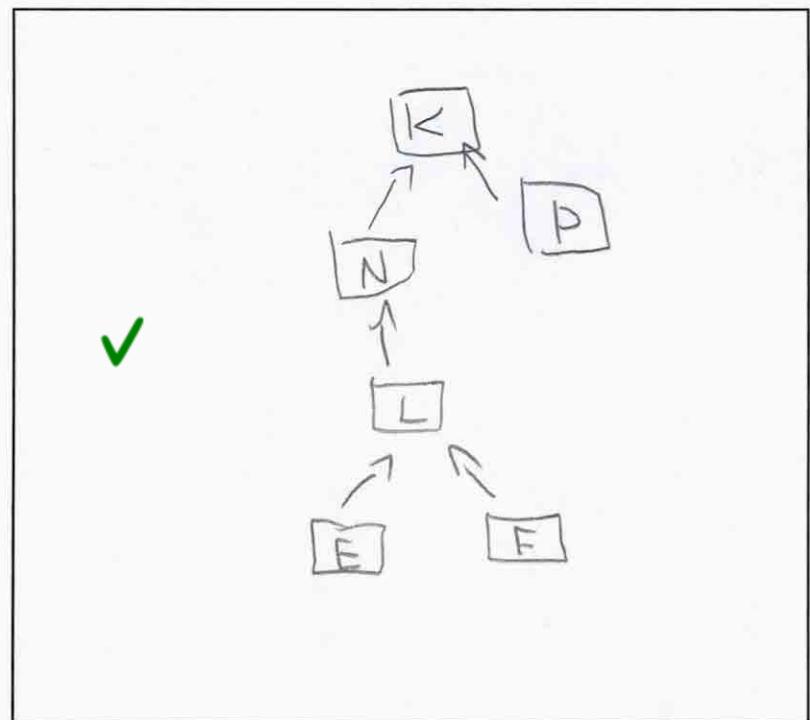
כמו כן, באותה מחלוקת Driver בהמשך השיטה main נכתבו פקודות נוספות (כל פקודה לא תלויה בפקודות האחרות), בהערה בסוף כל שורה, כתוב מה קרה בעקבות הוספה פקודה זו:

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

סעיף א (6 נקודות)



צירוף עץ ירושה אפשרי של כל המחלקות:
לדוגמא, אם המחלקה B יורשת שירות מהמחלקה A, צירוף זאת כך:
צירוף CAN:



סעיף ב (6 נקודות)

כתבו באילו מחלקות מופיעה השיטה ()f.

בכל מחלוקת כתבו את המימוש של השיטה ()f.

התשובה היא:

D:

```

public static String f()
{
    return "f1";
}
  
```

```

L: public static String f()
{
    return "f3";
}
  
```

```

F: public static String f()
{
    return "f2";
}
  
```

✓

סעיף ג (6 נקודות)

במהלך 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, סמן בצדקה ברורה ומפורשת אם היא תקינה או גורמת לשגיאת קומpileציה או גורמת לשגיאת ריצה.

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

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

הפקודות אינן תלויות זו בזו.

a) `arr[3] = lf;`

תיקין / שגיאת קומpileציה / שגיאת זמן ריצה

הסבר: לפניהם אין ערך קבוע נוגע לאפשרות מילוי
ללא ערך קבוע כיוון שהוא מופיע לפני כל אחד

b) `lf = arr[3];`

תיקין / שגיאת קומpileציה / שגיאת זמן ריצה

הסבר: אלאו אין ערך קבוע נוגע לאפשרות מילוי
ללא ערך קבוע (7) אונליין (1)

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

תיקין / שגיאת קומpileציה / שגיאת זמן ריצה

הסבר: true
כל אובייקט הוא אובייקט של אובייקט

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

תקין / שגיאת קומpileציה / שגיאת זמן ריצה

true

הסבר:



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

תקין / שגיאת קומpileציה / שגיאת זמן ריצה

false

הסבר:



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

תקין / שגיאת קומpileציה / שגיאת זמן ריצה

הסבר: ~~הונכיה "o1" איננה עוזרת בבדיקה האם o1 הוא Object או לא. בדיקת instanceof מושגת באמצעות o1 instanceof E~~

~~הונכיה "o1" איננה עוזרת בבדיקה האם o1 הוא Object או לא. בדיקת instanceof מושגת באמצעות o1 instanceof E~~

18
(6)



בהתכלחה!

