

## חלק א – עליכם לענות על כל השאלות בחלק זה במחברת הבחינה

### שאלה 1 (25 נקודות)

הנסיך מקפץ על גגות העיר. מטרתו למצוא את הרשע ולהילחם בו, על מנת להציל את הנסיכה. מפת הגבהים של גגות העיר מיוצגת על-ידי מערך דו-ממדי ריבועי המכיל מספרים שלמים. הניחו כי אם התא מכיל מספר אי-שלילי, הוא מייצג גובה של גג. המיקום של הרשע מסומן על-ידי המספר 1-.

בכל צעד הנסיך יכול להתקדם למשבצת סמוכה: צפונה, דרומה, מזרחה או מערבה (לא באלכסון). אם המשבצת הסמוכה נמצאת באותו גובה – יכול הנסיך ללכת אליה. בנוסף, יכול הנסיך לטפס על גג בגובה יחידה אחת, או לרדת מגג בגובה יחידה אחת או שתיים. אם הוא מנסה לטפס לגובה של יותר מיחידה אחת או לרדת לגובה של יותר משתי יחידות, הוא נפסל מיד. כשהנסיך נמצא על גג סמוך לרשע (אחד מארבעת שכניו) הוא יכול לקפוץ אליו ללא קשר להפרש הגבהים בינו לבין הרשע.

עליכם לכתוב שיטה סטטית רקורסיבית שתתכן את המסלול שיביא את הנסיך לרשע במספר המשבצות הנמוך ביותר בלי להיפסל.

אפשר להניח כי המערך מייצג בצורה נכונה את הגבהים של גגות העיר. כל הערכים בתאים הם מספרים אי-שליליים ויש רק תא אחד שמחזיק מספר שלילי והוא 1- . אין תאים נוספים במערך שיש בהם מספרים שליליים.

**חתימת השיטה היא:**

```
public static int prince(int[] [] drm, int i, int j)
```

כאשר מפת הגבהים של גגות העיר נתונה על-ידי הפרמטר `drm` (Digital Roof Map) ואילו `i` ו-`j` מציינים את אינדקס השורה והעמודה בהתאמה של התא בו מתחיל הנסיך.

על השיטה להחזיר את מספר התאים במסלול הקצר ביותר או 1- אם אין מסלול חוקי כזה. לאחר ריצת השיטה על המפה (המערך) להישאר ללא שינוי.

לדוגמא, עבור המפה הבאה:

	0	1	2	3	4
0	2	0	1	2	3
1	2	3	5	5	4
2	8	-1	6	8	7
3	3	4	7	2	4
4	2	4	3	1	2

הרשע נמצא בתא בשורה 2 בעמודה 1.

אם הנסיך נמצא בתא (0,0) קיימים שלושה מסלולים המובילים אותו לרשע. הם מסומנים במפות לחלן:

2	0	1	2	3
2	3	5	5	4
8	-1	6	8	7
3	4	7	2	4
2	4	3	1	2

מסלול באורך 10

2	0	1	2	3
2	3	5	5	4
8	-1	6	8	7
3	4	7	2	4
2	4	3	1	2

מסלול באורך 10

2	0	1	2	3
2	3	5	5	4
8	-1	6	8	7
3	4	7	2	4
2	4	3	1	2

מסלול באורך 4

חשוקה קצאוציפ 8-9 קאחזרת חשוקה

לכן השיטה ותחזיר 4.

באותה מפה, אם הנסיך נמצא בתא (4,4) אין מסלול חוקי בו הוא יכול ללכת עד הרשע ולכן השיטה תחזיר 1-.

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

אפשר להשתמש בהעמסת-יתר (overloading).

מותר לשנות את המערך drin במהלך השיטה, אבל חובה להחזיר אותו למצבו ההתחלתי בסופה.

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)

נתון מערך מלא במספרים שלמים, שבו כל מספר מופיע פעמיים ברצף פרט למספר אחד שמופיע רק פעם אחת. המערך אינו ממוין. לדוגמא, המערכים הבאים מקיימים את התנאי:

	0	1	2	3	4	5	6	7	8
a =	6	6	18	18	-4	-4	12	9	9

	0	1	2	3	4	5	6	7	8	9	10	11	12
b =	8	8	-7	-7	3	3	0	0	10	10	5	5	4

	0
c =	5

כתבו שיטה סטטית שמקבלת כפרמטר מערך שמקיים את התנאי הנ"ל, ומחזירה את המספר שמופיע במערך רק פעם אחת. לדוגמא, במערכים לעיל:

תשובה 11  
לאחרת תשובה

- המספר הבודד במערך a הוא 12 שנמצא באינדקס 6
- המספר הבודד במערך b הוא 4 שנמצא באינדקס 12
- המספר הבודד במערך c הוא 5 שנמצא באינדקס 0

אתם יכולים להניח שהמערך אינו ריק ושהוא מקיים את התנאי, אין צורך לבדוק זאת.

חתימת השיטה היא:

```
public static int findSingle (int [] a)
```

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.

מה סיבוכיות זמן הריצה של השיטה שכתבתם? הסבירו תשובתכם.

אל תשכחו לתעד את מה שכתבתם!

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

**שאלה 3 (16 נקודות)**

נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

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

    public Node (int data)
    {
        _data = data;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getData()           {return _data; }
    public Node getLeftSon()        {return _leftSon; }
    public Node getRightSon()       {return _rightSon; }
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.  
בין השיטות נתונה השיטה what הבאה המקבלת כפרמטר את t מטיפוס Node שמצביע לשורש של  
עץ בינרי:

```
public static void what(Node t)
{
    what(t, "");
}

private static void what (Node t, String st)
{
    System.out.println (st);
    if (t.getLeftSon() != null)
        what ( t.getLeftSon() ,st + "0");
    if (t.getRightSon() != null)
        what (t.getRightSon(), st + "1");
}
```

**ענו על הסעיפים הבאים:**

1. לשיטה what הועברו כפרמטר שורש של עץ בינרי בשם root. הפלט שהתקבל לאחר הרצת השיטה היה (בסדר זה בדיוק):

0  
1  
10  
100  
101  
1011  
11  
110

ציירו את העץ הבינרי ששורשו root שהועבר כפרמטר לשיטה. יש לצייר את מבנה העץ. אם לדעתכם אי אפשר לצייר עץ לפי ההדפסה לעיל, הסבירו מדוע אי אפשר.  
**התשובה היא: (6 נקודות)**



2. לשיטה what הועבר כפרמטר שורש של עץ בינרי בשם root. הפלט שהתקבל לאחר הרצת השיטה היה (בסדר זה בדיוק):

0  
01  
011  
0110  
01100

ציירו את העץ הבינרי ששורשו root שהועבר כפרמטר לשיטה. יש לצייר את מבנה העץ. אם לדעתכם אי אפשר לצייר עץ לפי ההדפסה לעיל, הסבירו מדוע אי אפשר.  
**התשובה היא: (6 נקודות)**



3. לאחר הקלדת השאלה, נניח כי נפלה טעות בפלט שניתן בסעיף 2, ויש עוד שורה ובה מודפס המספר 1. איפה בפלט צריכה להיכנס שורה זו? האם היא יכולה להיכנס בכל מקום בפלט או במיקום מסוים? הסבירו ונמקו את תשובתכם.  
התשובה היא: (4 נקודות)

שאלה 4 (16 נקודות)

בחינת המחלקות AA, BB, ו-Run הבאות (כל מחלקה נמצאת בקובץ נפרד):

```
public class AA {
    private int _num1=0;
    private int _num2=0;
    protected static int count =0;

    public AA(int num)
    {
        _num1 = num;
        _num2 = num;
        count++;
        System.out.println ("AA ctor1");
    }

    public AA(int num1, int num2)
    {
        _num1 = num1;
        _num2 = num2;
        count++;
        System.out.println ("AA ctor2");
    }

    public int sum()
    {
        return _num1+_num2;
    }

    public void setNum(int num)
    {
        _num2 = num;
    }

    public static int getCount()
    {
        return count;
    }
}
```

```

public class BB extends AA {
    private int _num3 = 0;

    public BB(int num) {
        super (num);
        _num3 = num;
        System.out.println ("BB ctor1");
    }

    public BB(int num1, int num2, int num3) {
        super(num1, num2);
        _num3 = num3;
        System.out.println ("BB ctor2");
    }

    public int sum() {
        return super.sum()+_num3;
    }

    public void setNum(int num) {
        _num3 = num;
    }

    public static int getCount() {
        return count;
    }
} // end of class BB

```

//-----

```

public class Run {
    public static void main (String [] args) {
(1)        ✓ AA f1 = new AA(10);
(2)        ✓ System.out.println ("sum1= " + f1.sum());
(3)        ✓ System.out.println ("count= " + AA.getCount());
(4)        ✓ AA f2 = new AA (10, 20);
(5)        ✓ System.out.println ("count= " + AA.getCount());
(6)        ✓ BB s1 = new BB(1);
(7)        ✓ AA f3 = new BB(2);
(8)        ✓ System.out.println ("count= " + AA.getCount());
(9)        ✓ f2 = s1;
(10)       ✓ System.out.println ("sum= " + f2.sum());
(11)       ✓ s1.setNum(2);
(12)       ✓ System.out.println ("sum= " + s1.sum());
(13)       ✓ System.out.println ("sum= " + f2.sum());
(14)       f2.setNum(4);
(15)       System.out.println ("sum= " + f2.sum());
(16)       f1.setNum(4);
(17)       System.out.println ("sum= " + f1.sum());
(18)       System.out.println ("count= " + BB.getCount());
    }
}

```

count  
1

2

3

4

ON  
LE

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



## שאלה 5 (18 נקודות)

ברצוננו לאחסן טקסט במבנה נתונים.

נאחסן את הטקסט לא כסדרו, אלא כל מילה בטקסט תאוחסן עם הריבוי שלה (כלומר, מספר הפעמים בהם המילה מופיעה בטקסט). המילים יאוחסנו בצורה ממוינת לפי סדר הא"ב.

בשאלה זו נתייחס למילים המכילות אותיות קטנות בא"ב הלטיני בלבד.

נאחסן את הטקסט ברשימה מקושרת חד-סטרית. כל מילה תהיה בחוליה שתכיל אותה ואת הריבוי שלה.

כך נגדיר את המחלקה MultiStringNode המייצגת חוליה ברשימה:

```
public class MultiStringNode
{
    private String _element;
    private int _cardinality;
    private MultiStringNode _next;

    public MultiStringNode(String s)
    {
        _element = s;
        _cardinality = 1;
        _next = null;
    }

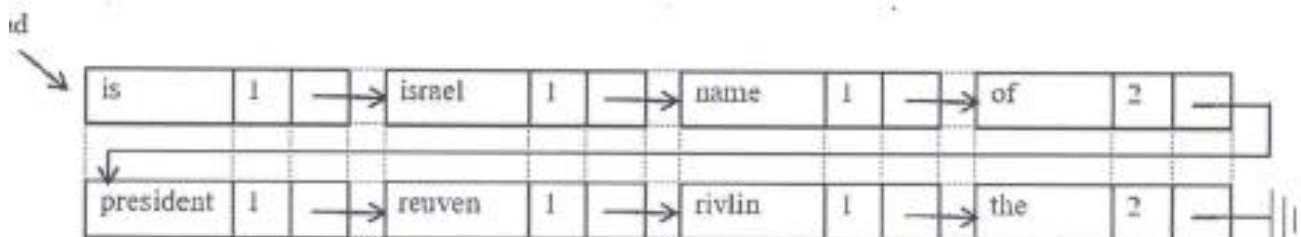
    public String getElement() { return _element; }
    public int getCardinality() { return _cardinality; }
    public MultiStringNode getNext() { return _next; }
    public void setElement(String s) { _element = s; }
    public void setCardinality(int num) { _cardinality = num; }
    public void setNext (MultiStringNode next) { _next = next; }
}
```

המחלקה MultiStringSet להלן מייצגת טקסט.

למשל, הטקסט הבא

the name of the president of israel is reuven rivlin.

יוצג על-ידי רשימה מקושרת בצורה הבאה (שימו לב שהרשימה ממוינת לפי המילים בסדר א"ב):



{(is, 1), (israel, 1), (name, 1), (of, 2), (president, 1), (reuven, 1), (rivlin, 1), (the, 2)}

במחלקה הוגדר בנאי ברירת-מחדל, וכן שיטה בשם add המקבלת מילה ומוסיפה אותה לרשימה

להלן קוד חלקי של המחלקה MultiStringSet

```
public class MultiStringSet
{
    private MultiStringNode _head;

    public MultiStringSet( ) {
        _head = null;
    }

    public MultiStringSet(MultiStringNode node) {
        _head = node;
    }

    public MultiNode add (String s) {...} //אין צורך לממש את השיטה

    public int f()
    {
        MultiStringNode temp = _head;
        int count = 0;
        while (temp != null)
        {
            count++;
            temp = temp.getNext();
        }
        return count;
    }

    public int g()
    {
        MultiStringNode temp = _head;
        int count = 0;
        while (temp != null)
        {
            count += temp.getCardinality();
            temp = temp.getNext();
        }
        return count;
    }
}
```

```

public MultiStringSet what (MultiStringSet s)
{
    MultiStringNode temp1 = _head;
    MultiStringNode temp2 = s._head;
    if (temp1 == null)
        return s;
    if (temp2 == null)
        return this;
    MultiStringSet newSet = new MultiStringSet();
    while (temp1!=null && temp2!=null)
    {
        if (temp1.getElement().compareTo(temp2.getElement()) < 0)
        {
            for (int i=0; i<temp1.getCardinality(); i++)
                newSet._head = newSet.add(temp1.getElement());
            temp1 = temp1.getNext();
        }
        else
            if (temp1.getElement().compareTo(temp2.getElement()) > 0)
            {
                for (int i=0; i<temp2.getCardinality(); i++)
                    newSet._head = newSet.add(temp2.getElement());
                temp2 = temp2.getNext();
            }
        else
        {
            int size = temp1.getCardinality()+
                temp2.getCardinality();
            for (int i=0; i<size; i++)
                newSet._head = newSet.add(temp2.getElement());
            temp1 = temp1.getNext();
            temp2 = temp2.getNext();
        }
    }
    while (temp2 !=null)
    {
        for (int i=0; i<temp2.getCardinality(); i++)
            newSet._head = newSet.add(temp2.getElement());
        temp2 = temp2.getNext();
    }
    while (temp1 !=null)
    {
        for (int i=0; i<temp1.getCardinality(); i++)
            newSet._head = newSet.add(temp1.getElement());
        temp1 = temp1.getNext();
    }
    return newSet;
}

} // end of class MultiStringSet

```

1. אם הטקסט הוא: `my name is avraham and her name is sarah` איזה ערך יוחזר אם נפעיל את השיטה `f` על אובייקט מהמחלקה `MultiStringSet` שמייצג את הטקסט?  
 התשובה היא: (1 נקודות)

---

2. אם הטקסט הוא: `my name is avraham and her name is sarah` איזה ערך יוחזר אם נפעיל את השיטה `g` על אובייקט מהמחלקה `MultiStringSet` שמייצג את הטקסט?  
 התשובה היא: (1 נקודות)

---

3. מה מבצעת השיטה `f` כשהיא מופעלת על אובייקט מהמחלקה `MultiStringSet` המייצג טקסט כלשהו? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת.  
 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, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה `f`, כשהיא מופעלת על אובייקט מהמחלקה `MultiStringSet` המייצג טקסט כלשהו. התייחסו למקרי קצה.  
 התשובה היא: (4 נקודות)

4. מה מבצעת השיטה `g` כשהיא מופעלת על אובייקט מהמחלקה `MultiStringSet` המייצג טקסט כלשהו? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת.  
 יציאו לך, וקליקת לחץ חמור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה `g`, כשהיא מופעלת על אובייקט מהמחלקה `MultiStringSet` המייצג טקסט כלשהו. התייחסו למקרי קצה. התייחסו להבדל בין `g` ל-`f` שלעיל.  
 התשובה היא: (4 נקודות)

1. אם נתון טקסט t1 שהרשימה המייצגת אותו היא :

{(abc, 3), (abd, 1), (bbc, 2), (ccc, 1), (cd, 1)}

וטקסט t2 שהרשימה המייצגת אותו היא :

{(aba, 4), (abd, 2), (add, 2), (cd, 2), (dd, 1), (ee, 1)}

מה יוחזר מהקריאה לשיטה s1.what(s2)

התשובה היא: (4 נקודות)

2. מה מבצעת השיטה what כשהיא מופעלת על אובייקט מחמלקה MultiStringSet המייצג

טקסט כלשהו text1 ומקבלת פרמטר אובייקט מחמלקה MultiStringSet המייצג טקסט

כלשהו text2? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת.

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, עליכם לכתוב מה המשמעות של הערך

שמוחזר מהשיטה what. התייחסו למקרי קצה.

התשובה היא: (6 נקודות)

בהצלחה!