

## מבוא לחישוב 2-7015710 סמסטר א'

קבוצות: 1,4,5,7,8,9

מבחן מועד ב' - 26.2.20

ליעד גוטליב, רויטל מרבל, סעיד עסלי.

- משך המבחן: 2.5 שעות.
- מחברת שורות. אין שימוש בחומר עזר.
- יש להחזיר את דף המבחן בסוף המבחן.
- במבחן חמש שאלות, כולם חובה.
- בכל שאלה ניתן לכתוב פונקציות עזר כרצונכם.

### שאלה 1 (20 נקודות)

מספר נקרא חצי-ראשוני (semi-prime) אם הוא תוצאת מכפלה של שני מספרים ראשוניים בלבד. כתבו פונקציה שמקבלת מספר שלם  $n$  ומחזירה `true` אך ורק אם הוא חצי-ראשוני.

```
public static boolean semi(int n)
```

דוגמאות:

המספר 4 הוא חצי-ראשוני: $2*2=4$	המספר 5 אינו חצי-ראשוני.
המספר 6 הוא חצי-ראשוני: $2*3=6$	המספר 12 אינו חצי-ראשוני: $2*2*3 = 12$
המספר 15 הוא חצי-ראשוני: $3*5=15$	המספר 30 אינו חצי-ראשוני: $5*3*2=30$

### שאלה 2 (20 נקודות)

נתונה מחרוזת המכילה תווים שונים ומספרים. כתבו פונקציה שמקבלת את המחרוזת, ומחזירה את המספר הגבוה ביותר שנמצא במחרוזת. ניתן להניח שאם יש מספרים במחרוזת הם חיוביים. במידה ואין שום מספר יש להחזיר -1. חתימת הפונקציה:

```
public static int findMaxNum(String str)
```

דוגמאות ריצה:

`findMaxNum("93&*ab1234crt70")` → 1234

`findMaxNum("8ab(55c#r9t")` → 55

`findMaxNum("hello")` → -1

### שאלה 3 (20 נקודות)

מטריצה נקראת "אלכסונית ממש" אם לכל אלכסון שיוורד משמאל לימין כל האיברים זהים. כתבו פונקציה שמקבלת מערך, ומחזירה true אם המערך מיצג מטריצה אלכסונית ממש, ואחרת מחזירה false. ניתן להניח שהמערך מיצג מטריצה, אבל לא בהכרח מטריצה ריבועית.

`public static boolean diagonal(int[][] a)`

דוגמה למטריצה אלכסונית ממש:  $\{1,2,3,4\}$ ,  $\{5,1,2,3\}$ ,  $\{9,5,1,2\}$ .

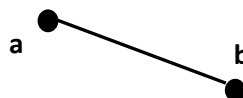
דוגמה למטריצה שאינה אלכסונית ממש:  $\{1,2\}$ ,  $\{2,2\}$ .

1	2
2	2

1	2	3	4
5	1	2	3
9	5	1	2

### שאלה 4 (20 נקודות)

בשאלה זו נבנה מחלקה בשם קטע (Segment). קטע במישור מוגדר על ידי שתי נקודות a, b. (ההגדרה של נקודה נמצאת בנספח, ויש להתייחס רק אליה ללא שינויים.)



1. כתבו את כותרת המחלקה ואת השדות (המשתנים) שלה.
2. כתבו שני בנאים: בנאי שמקבל את שני הנתונים, ובנאי מעתיק שמקבל קטע. בנאי המעתיק יעשה העתקה עמוקה.
3. כתבו שיטה שמחשבת ומחזירה את אורך הקטע.
4. כתבו שיטה שמקבלת נקודה חדשה, ומחזירה true אם הנקודה נמצאת על הקטע, ואחרת מחזירה false. (נקודה נמצאת על הקטע אם הסכום של המרחק מהנקודה ל-a והמרחק מהנקודה ל-b שווה לאורך הקטע.)

## שאלה 5 (20 נקודות)

רשימה מקושרת **לולאית** היא רשימה שהצומת (כלומר ה-node) האחרון מצביע על אחד מצמתי הרשימה שלפניה – ובכך אין צומת שמצביע על null. כתבו פונקציה שמקבלת צומת שהוא ראש של רשימה מקושרת, בודקת אם הרשימה היא לולאית, ואם כן מחזירה את האיבר הראשון בלולאה. אם הרשימה אינה לולאית, הפונקציה תחזיר null.

לדוגמה, עבור הרשימה הבאה:



הפונקציה תחזיר את הצומת שמכיל את הערך -30. עבור הרשימה הבאה:



הפונקציה תחזיר null.

```
public static Node cycle(Node p)
```

הדרכה: יש להתייחס למחלקה Node בנספח בלבד ללא שינויים. שימו לב שיתכן ששני צמתים שונים יאכסנו איברים שווים (כמו בדוגמה הראשונה בה המספר 3 מופיע פעמיים). לכן בדיקה ללולאה לא יכולה להתבסס על ערכי האיברים שברשימה.

```
public class Point {
    private double xVal;
    private double yVal;

    public Point(double x, double y) {
        xVal = x;
        yVal = y;
    }

    public double getX() { return xVal; }

    public double getY() { return yVal; }
}
```

```
public class Node {
    private int data;
    private Node next;

    public Node(int data) {
        this.data = data;
    }

    public int getData() {
        return data;
    }

    public Node getNext() {
        return next;
    }

    public void setNext(Node next) {
        this.next = next;
    }
}
```