

## מבוא לחישוב 2-7015710, סמסטר א'

### מבחן מועד א' - 30.1.18

אליזבת איצקוביץ, ליעד גוטליב, נועם חזון, 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.

#### שאלה 1 (20 נקודות)

מספר נקרא "מספר סכום ריבועי", אם הסכום של המחלקים שלו בריבוע הוא ריבוע. לדוגמה, המחלקים של 246 הם 1,2,3,6,41,82,123,246. סכום של המחלקים האלו בריבוע הוא

$$1+4+9+36+1681+6724+15129+60516 = 84100 = 290^2$$

לכן 246 נקרא מספר סכום ריבועי. כתבו פונקציה המקבלת מספר, ומחזירה **true** אם המספר הוא מספר סכום ריבועי. אחרת הפונקציה תחזיר **false**.

```
public boolean square(int a)
```

#### שאלה 2 (20 נקודות)

כתבו פונקציה בשם `reduce` שמקבלת מחרוזת ומחזירה אותה "מצומצמת" – כל רצף של אותיות זהות הופך לאות אחת.

```
public String reduce(String s)
```

דוגמה. קלט: "aaabbccccxxxzyza"

פלט: "abcxyza"

דוגמה. קלט: "abcdeb"

פלט: "abcdeb"

### שאלה 3 (20 נקודות)

כתבו פונקציה המקבלת שני מערכים דו-מימדיים של `int`, ומחזירה `true` במידה ואחד הוא שיקוף של השני על ציר ה-X או ה-Y. אחרת הפונקציה תחזיר `false`.

**`public boolean symmetric(int[][] a, int[][] b)`**

דוגמה לשיקוף על ציר ה-X:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 \end{pmatrix} \leftrightarrow \begin{pmatrix} 17 & 18 & 19 \\ 13 & 14 & 15 & 16 \\ 8 & 9 & 10 & 11 & 12 \\ 5 & 6 & 7 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

דוגמה לשיקוף על ציר ה-Y:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 \end{pmatrix} \leftrightarrow \begin{pmatrix} 4 & 3 & 2 & 1 \\ 7 & 6 & 5 \\ 12 & 11 & 10 & 9 & 8 \\ 16 & 15 & 14 & 13 \\ 19 & 18 & 17 \end{pmatrix}$$

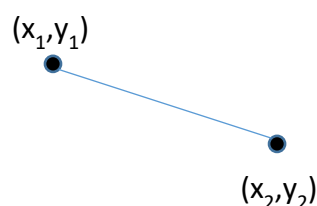
### שאלה 4 (20 נקודות)

- כתוב מחלקה בשם `line` המייצגת קטע של קו. למחלקה ארבעה שדות, כולם מספרים שלמים: `x, y` של נקודת סוף הראשונה, ו-`x, y` של נקודת סוף השנייה. (המחלקה אינה מכילה נקודות, רק מספרים שלמים.)
- הוסף למחלקה שני בנאים: בנאי המקבל ארבעה פרמטרים, ובנאי מעתיק.
- הוסף למחלקה פונקציה המחזירה את אורך הקטע.

**`public double length()`**

- הוסף למחלקה פונקציה שמקבלת נקודה `(x, y)`, ומחזירה `true` אם הנקודה נמצאת על הקטע.

**`public boolean on(int x, int y)`**



## שאלה 5 (20 נקודות)

מצורפת למבחן מחלקת **MyLinkedList** המייצגת רשימה מקושרת חד-כיוונית. יש להוסיף למחלקה מתודה (פונקציה)

**public boolean swap(int i, int j)**

המתודה מקבלת שני אינדקסים  $i, j$  ומחליפה בין שני האיברים של הרשימה, כלומר איבר הנמצא במקום  $i$  הולך למקום  $j$  ואיבר הנמצא במקום  $j$  הולך למקום  $i$ . המתודה מחזירה **true** אם ההחלפה הצליחה. במקרה שהאינדקסים חורגים את גודל הרשימה או קטנים מאפס היא מחזירה **false**.

### דוגמה:

א) רשימה:  $[a, b, c, d, e, f]$

קלט:  $i=1, j=5$

רשימה:  $[a, f, c, d, e, b]$

פלט: **true**

ב) רשימה:  $[a, b, c, d, e, f]$

קלט: הרשימה  $i=1, j=6$

רשימה:  $[a, b, c, d, e, f]$

פלט: **false**

## נספח: רשימה מקושרת:

```
public class Node {
    String data;
    Node next;
    public Node(String data){
        this.data = data;
        this.next = null;
    }
    public String toString(){
        return data;
    }
}

public class MyLinkedList {
    Node head;
    int size;

    public MyLinkedList(){
        head = null;
        size = 0;
    }

    public void add(String data){
        if (head == null){
            head = new Node(data);
        }
        else{
            Node n = head;
            while(n.next != null){
                n = n.next;
            }
            n.next = new Node(data);
        }
        size++;
    }

    public String toString(){
        String ans = "[";
        if (head == null) ans = "[]";
        else {
            Node n = head;
            while(n.next != null){
                ans = ans + n.data + ", ";
                n = n.next;
            }
            ans = ans + n.data + "];"
        }
        return ans;
    }
}
```