

מבוא לחישוב 3,8 - 2-7015710 סמסטר ב'

מבחן מועד א' – 16.06.21

גב' אליזבט איצקוביץ, גב' רויטל מרבל.

- משך המבחן: 2.5 שעות.
- מחברת משבצות. אין שימוש בחומר עזר.
- יש להחזיר את מחברת המבחן בסוף המבחן.
- במבחן חמש שאלות, כולם חובה.

שאלה 1 (20 נקודות)

(א 15 נק') פלינדרום (**palindrome**) הוא מילה, ביטוי, מספר או רצף אחרים של תווים שהקריאה שלו זהה מהסוף להתחלה או מהתחלה לסוף, כמו "madam" או "racecar"

כתבו פונקציה בוליאנית שמקבלת מחרוזת תווים ומחזירה true אם המחרוזת מהווה פלינדרום, אחרת היא מחזירה false.

```
public static boolean isPalindrome(String s){...}
```

(ב 5 נק') מהו הפלט של הביטוי הבא, נמקו היטב.

```
public static void main(String[] args) {  
    String s1 = new String("pankaj");  
    String s2 = new String("PANKAJ");  
    System.out.println(s1 == s2);  
}
```

שאלה 2 (20 נקודות)

בשאלה זו יש להוסיף למחלקת `MyLinkedList` שתי מתודות הבאות:

(א 5 נק') מתודה המחזירה את מספר איברי הרשימה:
`public static int getSize(){...}`

(ב 15 נק') מתודה המוחקת את האיבר האמצעי של הרשימה.
`public static void removeMiddle(){...}`

דוגמה: קלט, `1→2→3→4→5`; פלט: `1→2→4→5`;

קלט, `1→2→3→4→5→6`; פלט: `1→2→3→5→6`;

שאלה 3 (20 נק')

א) (10 נק') כתבו פונקציה הממיינת מערך המורכב מאפסים ואחדות בלבד. יש להשתמש בלולאה אחת בלבד. אין להשתמש במיונים כלשהם.

```
public static void sort01(int[] arr){...}
```

דוגמה: קלט: $\{1, 0, 1, 0, 0, 1, 1, 0\}$; arr = $\{0, 0, 0, 0, 1, 1, 1, 1\}$; פלט:

ב) (10 נק') מטריצה נקראת "ממוינת שורות" אם בכל שורה האיברים ממוינים בסדר עולה. מטריצה נקראת "ממוינת עמודות" אם בכל עמודה האיברים ממוינים בסדר עולה. מטריצה נקראת "ממוינת לחלוטין", אם בכל שורה ובכל עמודה האיברים ממוינים בסדר עולה. כתבו פונקציה בוליאנית, הפונקציה מחזירה true אם המטריצה ממוינת לחלוטין, אחרת היא מחזירה false.

```
public static boolean absoluteSorted(int[][] mat)
```

דוגמה: קלט: $\begin{pmatrix} -5 & 0 & 5 \\ -2 & 9 & 12 \\ 9 & 10 & 20 \\ 41 & 55 & 101 \end{pmatrix}$ פלט: true, קלט: $\begin{pmatrix} 1 & 3 & 7 \\ 0 & 5 & 4 \\ 8 & 2 & 6 \end{pmatrix}$ פלט: false.

שאלה 4 (20 נק')

כתבו מחלקה בשם Balloon המייצגת כדור במרחב. כדי להגדיר כדור במרחב צריך להגדיר קואורדינטות של נקודת מרכז הכדור עם שעורים ממשיים (x, y, z) ורדיוס (r) הכדור שהוא גם מספר ממשי.

א) (10 נק') יש להגדיר בנאי מחלקה המקבל שלושה שעורים של מרכז הכדור (x, y, z) ורדיוסו, ובנאי מעתיק.

ב) (4 נק') יש לכתוב מתודה המחשבת נפח הכדור. הנוסחה לחישוב נפח הכדור היא:

$$V = \frac{4}{3} \pi \cdot r^3$$

(r – הוא רדיוס הכדור).

```
public double volume(){...}
```

ג) (2 נק') יש לכתוב מתודה שמקבלת נקודה כלשהי במרחב ומחזירה true כאשר הנקודה נמצאת

בתוך הכדור, אחרת היא מחזירה false. הנקודה מיוצגת ע"י שלושה שעורים שלה: (x, y, z) .

```
public boolean isIn(double x, double y, double z){...}
```

ד) (2 נק') יש לכתוב מתודה שמשנה את רדיוס של הכדור:

```
public void setRadius(double radius){...}
```

ה) (2 נק') יש לכתוב מתודה שמחזירה רדיוס של הכדור:

```
public double getRadius(){...}
```

שאלה 5 (20 נקודות)

כתבו מחלקה בשם `BalloonContainer` המייצגת אוסף של כדורים על ידי מערך דינאמי בעל גודל משתנה לפי הצורך.

1. (4 נק') יש לכתוב הגדרת המחלקה (משתני המחלקה).

יש לכתוב בנאי מחלקה ללא ארגומנטים. (בנאי ריק)

יש לכתוב בנאי מעתיק למחלקה.

גודל התחלתי (`INIT_SIZE`) של מערך הכדורים הוא 3.

2. (4 נק') יש לכתוב מתודה שמוסיפה כדור חדש לאוסף:

```
public void add(Balloon b){...}
```

3. (5 נק') יש לכתוב מתודה המגדילה את מערך הכדורים:

```
private void resize(){...}
```

4. (5 נק') יש לכתוב מתודה המוחקת כל הכדורים שרדיוס שלהם קטן או שווה 1:

```
public void removeSmallest(){...}
```

5. (2 נק') יש לכתוב מתודה שמחזירה מספר כדורים הנמצאים בתוך הקונטיינר:

```
public int size(){...}
```

נספח קוד

נזכור מחלקת `String` שמייצגת מחרוזות, למחלקה יש מספר שיטות לרבות:

- `int length()` //מחזירה את אורך המחרוזת (מספר תווים)
- `String substring(int start, int end)` //מחזירה תת-מחרוזת `[start, end)`
- `char charAt(int i)` //מחזירה תו הנמצא במקום `i` של המחרוזת
- `int compareTo(String str)` //מחזירה +1 אם `s > str`
מחזירה -1 אם `s < str`
אחרת מחזירה אפס (שוויון מחרוזות).

רשימה מקושרת:

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

```

        return data;
    }
}
public class MyLinkedList {
    Node head;

    public MyLinkedList(){
        head = null;
    }

    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);
        }
    }

    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;
    }
}

```