עבודת הגשה גאווה

תרגיל 1:

כתוב פונקציה המקבלת מטריצה ריבועית (משמע, מספר השורות שווה למספר העמודות).

הפונקציה תחזיר TRUE אם סכום האיברים בכל שורה שווה לסכום האיברים בעמודה המתאימה, ותחזיר FALSE אחרת.

כלומר, יש לבדוק אם סכום איברי השורה הראשונה שווה לסכום העמודה הראשונה וכו'.

<u>דוגמא</u>: עבור המטריצה הבאה הפונקציה תחזיר TRUE, מאחר וסכום איברי השורה הראשונה הוא 8 וכנ"ל סכום איברי העמודה הראשונה. סכום איברי השורה השניה הוא 16 וכנ"ל סכום איברי העמודה השניה וסכום איברי השורה השלישית הוא 8 וכנ"ל סכום איברי העמודה השלישית:

2	1	5
6	7	3
0	8	0

תרגיל 2:

מספר קַפְּרֶקַר הוא מספר טבעי השווה לסכום הרישא והסיפא של הייצוג העשרוני של ריבועו. המספרים קרויים כך על-שם המתמטיקאי ההודי דאטאראייה רמאצ'אנדרה קפרקר [מתוך ויקיפדיה].

<u>דוגמאות:</u>

- 9 הוא מספר קפרקר, מכיוון שריבוע ספרותיו הוא 81 ($9^2 = 81$) ו- $9 = 1 + 8$ (הרישא היא 8 והסיפא היא 1).
- גם 95121 מקיים את אותה תכונה: $95121^2 = 9048004641$, ופיצול המספר לשני חלקים (רישא וסיפא) מניב: $95121 = 04641 + 90480$

להלן דוגמה לכל מספרי הקפרקר עד 10,000, חזקתם וחלוקתם לרישא ולסיפא:

```
*** 9  (9^2 = 81) 8 -- 1
*** 45  (45^2 = 2025) 20 -- 25
*** 55  (55^2 = 3025) 30 -- 25
*** 99  (99^2 = 9801) 98 -- 1
*** 297  (297^2 = 88209) 88 -- 209
*** 703  (703^2 = 494209) 494 -- 209
*** 999  (999^2 = 998001) 998 -- 1
*** 2223  (2223^2 = 4941729) 494 -- 1729
*** 2728  (2728^2 = 7441984) 744 -- 1984
*** 4879  (4879^2 = 23804641) 238 -- 4641
*** 4950  (4950^2 = 24502500) 2450 -- 2500
*** 5050  (5050^2 = 25502500) 2550 -- 2500
*** 5292  (5292^2 = 28005264) 28 -- 5264
*** 7272  (7272^2 = 52881984) 5288 -- 1984
*** 7777  (7777^2 = 60481729) 6048 -- 1729
*** 9999  (9999^2 = 99980001) 9998 -- 1
```

<u>שימו לב</u>: חלוקת המספר לרישא ולסיפא אינה בהכרח באמצע!

עליכם להדפיס את 20 מספרי קפרקר החל מהמספר 9.

תרגיל 3:

להלן 5 פונקציות.

עבור כל פונקציה יש לצרף עץ המתאר הרצה יבשה של הקוד עבור הערכים שמצויינים בהערה בשורת חתימת הפונקציה, וכן לתאר במשפט בודד מה הפונקציה עושה (לציין **מה** הפונקציה מבצעת **ולא איך**).

שימו לב: אין להריץ שאלה זו, אלא לפתור ולהגישה על נייר.

```
public static int foo(int x)  {  // run with x=529
        if (x < 10)
                return x;

        return foo(x/10) + x%10;
}

public static int goo(int x[], int s) { // run with x={12, 91, 28}, s=3
        if (s == 1)
                return x[s-1];

        return goo(x, s-1) + x[0];
}

public static void moo(int n) {  // run with n=4
        if (n == 0)
                return;

        moo(n-1);
        for (int i=0 ; i < n ; i++)
                System.out.print("*");
        System.out.println();
}

public static int koo(int x) { // run with x=529
        if (x < 10)
                return x;

        int k = koo((x/100)*10+x%10);
        return (k*10+(x%100)/10);
}


public static int doo(int x) { // run with x=1043
        if (x < 10)
                return 1;

        return 1 + doo(x/10);
}
```

תרגיל 4:

כתוב פונקציה רקורסיבית המקבלת מספר ומחזירה את מספר הספרות הזוגיות.

תרגיל 5:

כתוב פונקציה רקורסיבית המקבלת מספר ומחזירה true אם כל ספרותיו זוגיות, false אחרת.

תרגיל 6:

נגדיר "מספר מתחלף" כמספר בו כל זוג ספרות שכנות (או צמודות) הינו בעל זוגיות שונה.

כתוב פונקציה רקורסיבית המקבלת מספר שלם חיובי n ותחזיר true אם הוא "מספר מתחלף", אחרת תחזיר false.

<u>דוגמאות</u>:

עבור המספר 163458 יוחזר true כי ליד כל ספרה זוגית יש ספרה אי זוגית.

עבור המספר 1634589 יוחזר true כי ליד כל ספרה זוגית יש ספרה אי זוגית.

עבור המספר 163789 יוחזר false כי הספרות 3 ו- 7 צמודות ושתיהן אי-זוגיות.

תרגיל 7:

כתבו את הפונקציה הרקורסיבית switchLetters:
public c static void switchLetters(StringBuilder str, int indexBegin, int indexEnd)
הפונקציה מקבלת מחרוזת, אינדקס של האיבר הראשון ואינדקס של האיבר האחרון.
הפונקציה תהפוך המחרוזת מהסוף להתחלה.
דוגמא:
עבור המחרוזת: "abcde" והמספרים 0 ו- 4, בסיום הפונקציה המחרוזת תהפוך להיות "edcba".

תרגיל 8:

כתוב פונקציה רקורסיבית המקבלת מספר בבסיס 10 ומספר נוסף המייצג בסיס (שיכול להיות 2 או 8 בלבד).
הפונקציה מחזירה מספר חדש המייצג את המספר שהתקבל בבסיס שהתקבל.

תרגיל 9 בעמוד הבא...

תרגיל 9:

בשאלה זו נטפל באורחים המתאכסנים בבית מלון.

לצורך כך הגדירו את המחלקות הבאות:

```
public class Guest {
   private String name;
   private int  passportNumber; // ערך זה יכול להיות רק שלם חיובי
}


public class Room {
   private int    numOfBeds;     // ערך זה יכול להיות רק שלם בין 1-4
   private Guest[] allGuests;
}



public class Hotel {
   private int numOfUsedRooms;
   private Room[][] allRooms;
}
```

במחלקה Room ישנו שדה המעיד על מספר המיטות בחדר. לא ניתן לאחסן בחדר יותר אורחים ממספר המיטות.

במחלקה Hotel התכונה allRooms הינה מטריצה של חדרים, כך שכל שורה במטריצה מייצגת חדרים בקומה במלון. ניתן להניח שבכל הקומות מספר החדרים זהה, ניתן להניח כי מספר הקומות קטן מ- 10 ומספר החדרים בכל קומה קטן מ- 100.

א. כתוב למחלקה Hotel בנאי המקבל את מספר הקומות ומספר החדרים בכל קומה. הבנאי יגריל את מספר המיטות בכל חדר במלון (ערך בין 1-4) ויאתחל בהתאם את נתוני המלון. יש לכתוב לשאר המחלקות בנאים בהתאם.

ב. כתוב במחלקה Hotel מתודה המקבלת מערך של אורחים ומשבצת אותם בחדר פנוי במלון (כל האורחים צריכים להיות באותו חדר שמספר המיטות בו הוא לפחות כמספר האורחים). המתודה תחזיר מספר המייצג את החדר בו שובצו האורחים: המספר יהיה תלת ספרתי כאשר ספרתו הראשונה היא מספר הקומה ושתי הספרות הבאות הן מספר החדר. במידה ולא נמצא חדר מתאים, יוחזר הערך 1-.

<u>דוגמאות:</u>

- עבור אורחים ששובצו בקומה השלישית בחדר 25 יוחזר 325

- עבור אורחים ששובצו בקומה השניה בחדר 9 יוחזר 209

ג. כתוב במחלקה Hotel מתודה המקבלת מספר פספורט של אורח ותחזיר את מספר החדר בו הוא מתארח. במידה והאורח אינו מתארח במלון יוחזר 1-.

ד. כתוב במחלקה Hotel מתודה המחזירה את הקומה בה מספר החדרים הפנויים הגדול ביותר.

ה. כתוב main והגדר בו מלון, ולאחר מכן הצג תפריט החוזר על עצמו ובו האפשרויות הבאות:

1- קליטת נתוני אורחים ושיבוצם בחדר במלון. יש להציג באיזה חדר שובצו האורחים או האם אין חדר מתאים. ההודעה תוצג ב- main ולא דרך המתודה.

2- קליטת מספר פספורט של אורח והצגה באיזה חדר הוא מתאחסן. אם האורח אינו מתאחסן במלון יש להציג הודעה מתאימה. ההודעה תוצג ב- main ולא דרך המתודה.

3- הצג את נתוני המלון: עבור כל קומה יש להציג נתוני האורחים המתאחסנים בכל חדר.

4- הצג באיזו קומה יש את מספר החדרים הפנויים הגדול ביותר.

ו. עם היציאה מה- main יש לשמור את נתוני המלון לקובץ. בתחילת ה- main יש לשאול את המשתמש האם הוא מעוניין לטעון את נתוני המלון מקובץ קיים ואם כן לקרוא את הנתונים מהקובץ (ניתן לעשות סעיף זה לאחר למידת הפרק "עבודה עם קבצים").

בהצלחה!