

במערכת מחשוב של חברה למכירת מכוניות משומשות הוגדרה מחלקה בשם **Car** שיש לה שלוש תכונות:

licenseNum — מספר רישוי מטיפוס מחרוזת.

hadAccident — משתנה מטיפוס בוליאני המקבל true אם המכונית עברה תאונה, אחרת — מקבל false.

price — מחיר המכונית הוא מספר מטיפוס שלם.

הנח שהוגדרה למחלקה פעולה בונה המקבלת ערכים עבור כל אחת מן התכונות של המחלקה, ולכל תכונה הוגדרו

ב־ Java פעולות get ו־ set וב־ C# פעולות Get ו־ Set.

א. כתוב במחלקה **Car** פעולה פנימית בוליאנית בשם range ב־ Java או Range ב־ C#, המקבלת שני מספרים

מטיפוס שלם min ו־ max. הפעולה תחזיר true אם מחיר המכונית הוא בין min ל־ max (כולל),

אחרת תחזיר הפעולה false.

הנח ש־ max גדול מ־ min.

ב. נתונה המחלקה **AllCars** שיש לה שתי תכונות:

cars — מערך חד ממדי מטיפוס **Car**.

num — מספר המכוניות הנוכחי שיש בחברה, מטיפוס שלם.

לפניך הפעולה הבונה של המחלקה:

```
public AllCars (int max)
{
    this.cars = new Car [max];
    this.num = 0;
}
```

(1) כתוב במחלקה **AllCars** פעולה פנימית בוליאנית בשם addCar ב־ Java או AddCar ב־ C#, המקבלת

עצם מטיפוס **Car**. הפעולה מוסיפה אותו למערך המכוניות במקום הראשון הפנוי ומחזירה true.

אם המערך מלא תחזיר הפעולה false.

הנח שאין מקומות פנויים מהתחלת המערך ועד למספר num (לא כולל num).

(2) כתוב במחלקה **AllCars** פעולה פנימית בשם print ב־ Java או Print ב־ C#, המקבלת שני מספרים

min ו־ max מטיפוס שלם (max גדול מ־ min).

הפעולה תדפיס את מספר הרישוי של כל מכונית במערך cars שלא עברה תאונה ושמחירה נמצא בין

שני המספרים min ו־ max (כולל).

חובה להשתמש בפעולה שהוגדרה בסעיף א.

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות:  
תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש  
בפעולות נוספות, עליך לממש אותן.

ענה על שתיים מן השאלות 4-6 (לכל שאלה – 25 נקודות).

---