

מתכונת מספר 2

פרק ראשון (25 נקודות)

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.
לפותרים ב-Java: בכל שאלה שנדרשת בה קליטה, הנח שבתכנית כתובה ההוראה:
Scanner input = new Scanner (System.in);

ענה על שאלה 1 – חובה (10 נקודות)

- כתוב פעולה חיצונית בוליאנית **check** ב-Java או **Check** ב-C# שתקבל מערך חד-ממדי של מספרים ממשיים, מספר שלם חיובי (num) ומערך דו-ממדי של מספרים ממשיים. הפעולה תחזיר true אם ערך האיבר במערך החד-ממדי במקום num נמצא בשורה הראשונה במטריצה או בשורה האחרונה במטריצה, בכל מקרה אחר הפעולה תחזיר false.
אם המספר num חורג מגבולות המערך יש להחזיר false.

ענה על אחת מן השאלות 2-3 (לשאלה ← 15 נקודות)

- נתונה המחלקה המייצגת עיר – **City** שלה ארבע תכונות:
שם העיר – name מטיפוס מחרוזת.
שטח העיר – surface מטיפוס ממשי.
מספר תושבים נוכחי בעיר – population מטיפוס שלם.
האם העיר שוכן ליד חוף ים – haveBeach מטיפוס בוליאני.
לפניך ממשק המחלקה **City**.

תיאור הפעולה	כותרת הפעולה
פעולה בונה המקבלת את ערכי התכונות כפרמטר.	public City(string name, double surface, int population, bool haveBeach)
פעולה המחזירה את שטח העיר.	public double GetSurface()
פעולה המחזירה את מספר התושבים בעיר.	public int GetPopulation()
פעולה המחזירה את שם העיר.	public int GetName()
פעולה המחזירה true אם העיר שוכן ליד חוף ים ו-false אחרת.	public bool HaveBeach()

א. ממש את הפעולות המופיעות בממשק של המחלקה **City**.

- לקוח מתוך הספר: "בחינות מתכונת במדעי המחשב" -

- ב. כתוב פעולה חיצונית המקבלת מערך של ערים.
מבין הערים ששוכנות ליד חוף ים, הפעולה תחזיר את שם העיר שהצפיפות בה הנמוכה ביותר. הנח שיש לפחות עיר אחת השוכנת ליד חוף ים.
ג. לפניך מימוש חלקי של המחלקה **Country** המייצגת מדינה:

```
public class Country
{
    private City [ ] allCities; // array of cities.
    private int currentcities; // current cities in the array.
    public Country()
    {
        this.allCities = new City [1000];
        this.currentcities = 0;
    }
    public void AddCity(City c)
    { [ ... ] }
}
```

3. נתונה המחלקה יומן חודשי – **MonthDiary** אשר מיוצגת ע"י מערך דו-ממדי של מחרוזות. כל מחרוזות מייצגות את האירוע שיש באותו יום בחודש. הפעולה הבונה של המחלקה מגדירה יומן חודשי ריק בגודל 4x7. (הנח שבכל חודש 28 ימים).
כל שורה במערך הדו-ממדי מייצגת שבוע וכל עמודה מייצגת יום בשבוע.

א. עבור המחלקה **MonthDiary** השלם את:

- i. כותרת המחלקה
 - ii. תכונות המחלקה
 - iii. הפעולה הבונה של המחלקה המגדירה יומן חודשי ריק.
- נתונה המחלקה יומן שנתי – **YearDiary** אשר מיוצגת ע"י מערך בגודל 12 מטיפוס **MonthDiary**. הפעולה הבונה של המחלקה מגדירה יומן שנתי ריק.
כל תא במערך החד-ממדי מייצג חודש בשנה.

ב. עבור המחלקה **YearDiary** השלם את:

- i. כותרת המחלקה
- ii. תכונות המחלקה
- iii. הפעולה הבונה של המחלקה המגדירה יומן שנתי ריק.

- ג. כתוב פעולה המקבלת יומן שנתי yd מטיפוס **YearDiary** של שנת 2017 ומחרוזות המייצגות אירוע. הפעולה תחזיר את התאריך של אירוע זה.
אם אירוע זה לא קיים ביומן, הפעולה תחזיר את התאריך : 0/0/0.

- לקוח מתוך הספר: "בחינות מתכונת במדעי המחשב" -

פרק שני (50 נקודות)

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)

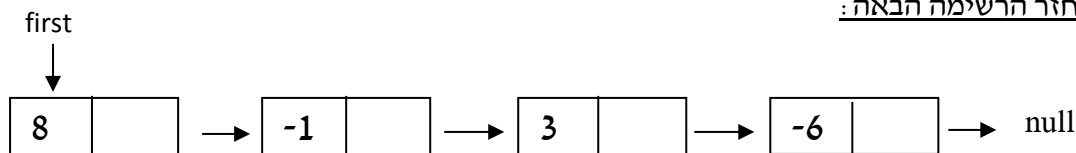
4. נגדיר "כתם" במחסנית כרצף של מספרים חיוביים או כרצף של מספרים שלילים במחסנית.

כתוב פעולה חיצונית המקבלת מחסנית של מספרים שלמים השונים מ-0. הפעולה תחזיר רשימה מקושרת כך שעבור כל "כתם" במחסנית תכנס לרשימה הסכום של אותו "כתם".
על המחסנית המקורית בסוף הפעולה להיות ללא שינוי.

לדוגמה, עבור המחסנית:



תוחזר הרשימה הבאה:



5. הפעולה **QueueCounter** מקבלת עץ בינארי **tr** המכיל מספרים שלמים, ומחזירה תור. בעבור כל מספר בעץ **tr**, יהיו בתור שני איברים: האיבר הראשון מכיל את המספר מהעץ **tr**, והאיבר השני מכיל את מספר הפעמים שהוא מופיע בעץ **tr**.
שים לב: בעבור מספר המופיע יותר מפעם אחת בעץ **tr**, יהיה זוג אחת בלבד בתור החדש.
הנחה: העץ **tr** לא ריק.

- א. השלם את הפעולה **QueueCounter**.
- ב. מהי סיבוכיות הפעולה **QueueCounter**? נמק את קביעתך.

- לקוח מתוך הספר: "בחינות מתכונת במדעי המחשב" -

6. עץ בינארי יקרא **מאוזן** אם עבור כל צומת בעץ, סכום אברי התת-עץ הימני של הצומת שווה לסכום אברי התת-עץ השמאלי של הצומת.

- א. ממש פעולה חיצונית Sum ב-C# או sum ב-Java. הפעולה תקבל עץ בינארי tr מטיפוס שלם. הפעולה תחזיר את סכום העץ.
- ב. ממש פעולה חיצונית Same ב-C# או same ב-Java. הפעולה תקבל עץ בינארי tr מטיפוס שלם ומספר שלם num. הפעולה תחזיר true אם סכום העץ שווה ל-num ו-false אחרת.
- ג. ממש את הפעולה המקבלת עץ בינארי מטיפוס שלם ומחזירה true אם העץ **מאוזן** ו-false אחרת.

כותרת המחלקה ב-C# : `public static bool Balance(BinNode<int> tr)`

כותרת המחלקה ב-Java : `public static boolean balance(BinNode<integer> tr)`

ד. כתוב קטע קוד במחלקה ראשית היוצר עץ בינארי **מאוזן** בעל תשעה צמתים.

פרק שלישי (25 נקודות)

בפרק זה שאלות במסלול מודלים חישוביים.

ענה על שאלה אחת במסלול זה.

מודלים חישוביים

אם למדת מסלול זה, ענה על אחת מן השאלות 7 – 8 (25 נקודות)

7. נתונה השפה L מעל ה- $\{a, b\}$:

$$L = \{ w_1 \cdot a^n \cdot w_2 \mid n = 1, |w_2| = 2 \}$$

w_1, w_2 - מילים מעל ה- $\{a, b\}$. ($w_1 \neq \varepsilon$).

א. הוכח שהשפה L היא שפה רגולרית ובנה אוטומט סופי דטרמיניסטי המקבל את שפה זו.

ב. על סרט זיכרון של מכונת טיורינג רשומה מילה מעל ה- $\{a, b\}$. בנה מכונת טיורינג הבודקת אם המילה הרשומה על הסרט שייכת לשפה L .

8. נתונות ארבע שפות מעל ה- $\{a, b\}$:

$$L_1 = \{ a^{n+1} b^{n+2} a^n \mid 0 \leq n \leq 1 \}$$

L_2 - שפה המכילה את כל המילים מעל ה- $\{a, b\}$ שיש בהם לפחות את אחד מן הרצפים הבאים:

- i. abb
- ii. bab
- iii. aba

$$L_3 = \overline{L_2} \cap R(L_1)$$

$$L_4 = L_1 \cap R(L_2)$$

א. בנה אוטומט סופי דטרמיניסטי מלא המקבל את השפה L_1 .

ב. רשום מילה השייכת לשפה L_3 . מה ניתן לומר על השפה הזאת?

ג. רשום מילה השייכת לשפה L_4 .

בהצלחה!