GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 2022 מועד א'
סמל השאלון: 97104
שם התלמיד/ה: _____

אלגוריתמיקה ותכנות (Java ו-C#)

הנדסת תוכנה · הנדסאים וטכנאים

הוראות

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: **בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:** מבחן ב- Java (עמוד 2) מבחן ב- C# (עמוד 15) בכל מבחן 14 שאלות. **חלק א' – 40 נקודות** שאלות 1-6: יש לענות על **ארבע** שאלות בלבד. ערך כל שאלה – 10 נקודות. **חלק ב' – 24 נקודות** שאלות 7-10: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 12 נקודות. **חלק ג' – 36 נקודות** שאלות 11-14: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 18 נקודות. **בסך הכול: 100 נקודות.**

ג. חומר עזר מותר בשימוש: 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית). 2. קלסר **אחד בלבד** עם **חומר ההרצאות**. אין להוציא דפים מהקלסר. **אין לצרף ספרים או חוברות עם פתרונות.**

ד. <u>הוראות מיוחדות</u>: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת הבחינה). 3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות. 4. יש לכתוב את התשובות במחברת הבחינה **בעט בלבד**, בכתב יד ברור. 5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה. 6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק. 7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד. 8. יש להסביר בפירוט כל תוכנית שנכתבה, **תוכנית ללא הסבר מפורט לא תזכה בניקוד.** 9. אם לדעתכם חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לכם להמשיך בפתרון השאלה. נמקו את בחירתכם. **הנחיות כלליות לנבחנים - מבחן ב-1 **JAVA:. בכל שאלה שנדרשת בה קליטה, אין צורך בבדיקת תקינות הקלט. 2. בכל שאלה שנדרשת בה קליטה, הנח שבתוכנית כתובה ההוראה: `Scanner in=new Scanner(System.in);` 3. דוגמה להוראה לקליטת מספר שלם: `int x = in.nextInt();` 4. דוגמה להוראה לקליטת מספר עשרוני: `double y = in.nextDouble();` 5. דוגמה להוראה לקליטת מחרוזת: `String str = in.next();` 6. הוראות לפלט על המסך: `System.out.print();` `System.out.println();`. יצירת מספר שלם אקראי num בין X ל- Y כולל (X <= num <= Y): שיטה א' - שימוש במחלקת שירות Math: `int num = X + (int)((Y - X +1)*Math.random());` שיטה ב' - שימוש במחלקה Random: `Random rand = new Random();` `int num = X + rand.nextInt(Y - X + 1);` **הנחיות כלליות לנבחנים - מבחן ב-1 **C#:. בכל שאלה שנדרשת בה קליטה, אין צורך בבדיקת תקינות הקלט. 2. דוגמה להוראה לקליטת מספר שלם: `int x = int.Parse(Console.ReadLine());` 3. דוגמה להוראה לקליטת מספר עשרוני: `double y = double.Parse(Console.ReadLine());` 4. דוגמה להוראה לקליטת מחרוזת: `string str = Console.ReadLine();` 5. הוראות לפלט על המסך: `Console.Write();` `Console.WriteLine();`. יצירת מספר שלם אקראי num בין X ל-Y כולל (X <= num <= Y): `Random rand = new Random();` `int num = X + rand.Next(X, Y + 1);`

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

1. כתבו קטע תוכנית הקולט **זוגות** מספרים שלמים. הקלט מסתיים כאשר ייקלט זוג מספרים נגדיים.

דוגמאות למספרים נגדיים: (2, -2), (-10, 10).

על התוכנית לחשב ולהדפיס:

- כמה מספרים נקלטו.
- סכום של כל המספרים החיוביים שנקלטו.
- כמה זוגות של מספרים שווים נקלטו.

2. כתבו פעולה אשר מקבלת מספר שלם וחיובי `num` ומחזירה מערך של מספרים שלמים בגודל `num` מלא במספרים **אקראיים דו-ספרתיים חיוביים** כך שבכל מספר **ספרת האחדות גדולה מספרת העשרות**.

יש לבצע את מילוי המערך בלולאה אחת בלבד!

3. מערך של מספרים שלמים נקרא "מערך משולש" אם הוא מורכב משלושה חלקים זהים.

לדוגמה:

- המערך `int[] a = {13,6,3,13,6,3,13,6,3}` הוא "מערך משולש".
- המערך `int[] b = {1,1,1,1,1,1}` הוא "מערך משולש".
- המערך `int[] c = {1,2,3,4,1,2,3,4,1,2,3}` הוא לא "מערך משולש".
- המערך `int[] d = {1,2,1,2,2,2}` הוא לא "מערך משולש".

א. כתבו פעולה המקבלת מערך של מספרים שלמים ובודקת אם הוא "מערך משולש".

אם כן - הפעולה תחזיר ערך `true`, ולא, הפעולה תחזיר ערך `false`.

(7 נקודות)

ב. מהי סיבוכיות הפעולה מסעיף א'? הסבירו את תשובתכם.

(3 נקודות)

4.

בבית קולנוע יש שישה אולמות, ממוספרים 1-6, בכל אולם יש 200 מקומות.
אולם מספר 1 נועד לסרטי ילדים, ואף פעם לא מתבטלת שם הקרנת סרט. בהפסקה מחלקים לכל ילד משולש פיצה וטרופית.
ביתר האולמות - אם יש פחות מעשרה צופים, לא מתקיימת הקרנת הסרט והצופים מועברים לאולמות אחרים.
אדם המגיע לקופה, מציין את מספר האולם, ואת כמות הכרטיסים שבה הוא מעוניין.
סיום קליטת הנתונים לערב יהיה קלט למספר **אולם** לא תקין.
הנהלת הקולנוע, מעוניינת לקבל בכל ערב (בסיום מכירת הכרטיסים) את המידע הבא:

- הצגה כפלט של האולמות שבהם **לא תיערך הקרנה**.
- הצגה כפלט את מספרי האולמות בהם יש מקומות פנויים, וכמה צופים אפשר להעביר אליהם.
- כמה **ארגזי טרופית** וכמה **פיצות** יש להזמין עבור אולם הילדים (בפיצה יש שמונה משולשים, ובארגז טרופית יש 20 טרופיות).

כתבו תוכנית אשר תקלוט נתוני הזמנות ותדפיס את המידע הדרוש להנהלה.

5. המחלקה שיר (`Song`) מאפיין שירים ומורכב משלוש תכונות:
- `name` - שם השיר, מטיפוס מחרוזת.
- `performer` - שם המבצע, מטיפוס מחרוזת.
- `length` - אורך השיר בשניות, מטיפוס מספר שלם.

במחלקה הוגדרו בנאי (פעולה בונה, constructor), פעולות set/get לכל התכונות ופעולה toString.
לאחר ביצוע קטע התוכנית הבא:

Java:

```
Song ns=new Song("Ququ",120,"Buku");
System.out.println(ns.status());
```

C#:

```
Song ns=new Song("Ququ",120,"Buku");
Console.WriteLine(ns.Status());
```

התקבל הפלט הבא:

```
Buku/Ququ:120
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

א. כתבו את בנאי המחלקה ואת הפעולה `status()` בהתאם לפלט שהתקבל.

(3 נקודות)

להלן המחלקה `Disc` המאפיינת דיסק עם שירים, למחלקה שתי התכונות הבאות:

- שם הדיסק (`discName`), מטיפוס מחרוזת.
- מערך שירים (מערך עצמים מטיפוס `Song`) `songs`.

במחלקה הוגדרו בנאי (פעולה בונה, constructor), פעולות set/get לכל התכונות ופעולה toString.

ב. כתבו במחלקה `Disc` את הפעולה:

Java:

```
public boolean  exist(String nSong, String pSong)
```

C#:

```
public bool  Exist(string nSong, string pSong)
```

הפעולה תבדוק אם שיר בעל שם `nSong` ושם המבצע `pSong` נמצא בדיסק.

אם כן, הפעולה תחזיר true, ולא הפעולה תחזיר false.

(3 נקודות)

ג. כתבו את הפעולה החיצונית:

Java:

```
public static String largeDiscName(Disc d1, Disc d2)
```

C#:

```
public static String largeDiscName(Disc d1, Disc d2)
```

הפעולה מקבלת שני דיסקים ומחזירה **שם של הדיסק** שיש בו יותר שירים.

(4 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

6. נתונה הפעולה what המקבלת מערך מספרים שלמים וחיוביים.

Java:

```
public static int what(int[] arr)
{
  int x = arr[0];
  int y = 0;
  int z = 0;
  for (int i = 1; i < arr.length; i++)
  {
    if (arr[i] > x)
    {
      z = y;
      y = x;
      x = arr[i];
    }
    else
     if (arr[i] > y)
     {
       z = y;
       y = arr[i];
     }
     else
      if (arr[i] > z)
      {
        z = arr[i];
      }
  }
  return x + y + z;
}
```

C#:

```
public static int What(int[] arr)
{
  int x = arr[0];
  int y = 0;
  int z = 0;
  for (int i = 1; i < arr.Length; i++)
  {
    if (arr[i] > x)
    {
      z = y;
      y = x;
```

```
        x = arr[i];
      }
      else
       if (arr[i] > y)
       {
         z = y;
         y = arr[i];
       }
       else
        if (arr[i] > z)
        {
          z = arr[i];
        }
   }
   return x + y + z;
}
```

א. עקבו בעזרת טבלת המעקב אחרי ביצוע הפעולה what עבור המערך:

`int[] arr={12, 13, 1, 10, 34}`

ורשמו מה תחזיר הפעולה.

(5 נקודות)

ב. מה מבצעת הפעולה what באופן כללי עבור המערך של מספרים שלמים?

(3 נקודות)

ג. מהי סיבוכיות של הפעולה? הסבירו את תשובתכם.

(2 נקודות)

7. נתונה המחלקה Student הבאה:

Java:

```
public class Student
{
   private String id; // תעודת זהות
   private int[] arrGradesSemesterA; // מערך ציוני סמסטר א'
   private int[] arrGradesSemesterB; // מערך ציוני סמסטר ב'
}
```

C#:

```
public class Student
{
   private string id; // תעודת זהות
   private int[] arrGradesSemesterA; // מערך ציוני סמסטר א'
   private int[] arrGradesSemesterB; // מערך ציוני סמסטר ב'
}
```

המחלקה מתארת את ציונים של סטודנט בסמסטר א' ובסמסטר ב'.

- התכונה id מחזיקה את תעודת הזהות של הסטודנט.
- המערכים arrGradesSemesterA ו- arrGradesSemesterB, מחזיקים את ציוני הסטודנט בסמסטר א' ו- ב' בהתאמה עבור הקורסים בסמסטר א' וסמסטר ב', שני המערכים באותו האורך.

אם מקצוע מסוים לא נלמד בסמסטר א' או בסמסטר ב', בתא המתאים במערך מופיע מספר -1, ולא מופיע ציון הגדול או שווה לאפס. במחלקה הוגדרו פעולות get/set לכל התכונות.

סטודנט נחשב "משתפר" אם הוא שיפר את הישגיו בסמסטר השני או לפחות **לא חלה ירידה** בשום מקצוע בסמסטר השני, כלומר מספיק מקצוע אחד שבו ירדו הישגיו בסמסטר ב' כדי לא להיחשב סטודנט משתפר.

אם מקצוע מסוים נלמד רק בסמסטר אחד, אין צורך בבדיקת שיפור במקצוע זה.

א. כתבו את הפעולה `public boolean isImproved()`.

הפעולה מחזירה true אם סטודנט הוא "סטודנט משתפר" ואם לא, הפעולה מחזירה false.

אם אין אף מקצוע שנלמד בשני סמסטרים, הפעולה תחזיר false.

לדוגמה: אם נתון מערך ציוני מחצית א' הבא:

מקום	0	1	2	3	4	5
ציון	60	50	80	-1	60	70

ומערך ציוני מחצית ב':

מקום	0	1	2	3	4	5
ציון	80	90	90	75	-1	70

הפעולה `isImproved` תחזיר true כיוון שעבור כל ציון במחצית א', הציון המתאים במחצית ב' לא היה נמוך יותר, ואילו היה מערך ציוני מחצית ב' הבא:

מקום	0	1	2	3	4	5
ציון	80	90	50	75	-1	70

הפעולה `isImproved` תחזיר false מכיוון שעבור המקצוע השלישי (מיקום 2) הציון בסמסטר ב' היה נמוך יותר מהציון של בסמסטר א'.

(6 נקודות)

ב. כתבו פעולה חיצונית אשר מקבלת מערך עצמים מסוג Student. הפעולה תחזיר מערך תעודות הזהות של כל הסטודנטים ה"משתפרים". אם אין אף סטודנט "משתפר", הפעולה תחזיר `null`.

(6 נקודות)

8. **מכרז "מחיר שני" (מכרז ויקרי)** VickeryAuction הוא סוג של מכירה פומבית במעטפות סגורות (כלומר, אף מתמודד לא רואה את ההצעות שהגישו מתמודדים אחרים). המנצח במכרז הוא מי שהגיש את ההצעה הגבוהה ביותר. לפי כללי ה"מכרז ויקרי", המנצח לא משלם את ההצעה שלו, אלא את ההצעה השנייה בגובהה.

נגדיר טיפוס **משתתף במכרז** – Participant המאופיין כך:

- שם המשתתף – name מטיפוס מחרוזת.
- תעודת זהות – id מטיפוס מחרוזת.
- המחיר אותו המשתתף מציע – price מטיפוס שלם.

הטיפוס **מכרז ויקרי** VickeryAuction מאופיין על-ידי:

- תיאור המוצר הנמכר – product מטיפוס מחרוזת.
- ערך ריאלי של המוצר – realPrice מטיפוס שלם.
- מערך arr בגודל 100 של משתתפים מטיפוס Participant.
- מספר המשתתפים שנרשמו בפועל – current מטיפוס שלם.

א. כתבו את כותרות המחלקות Participant ו- VickeryAuction ואת התכונות שלהן.

(3 נקודות)

משתתף יכול להתווסף למערך המשתתפים רק אם יש מקום פנוי וכן המחיר שאותו הוא מציע גבוה מ- 50% הערך הריאלי של המוצר.

ב. כתבו פעולה בוליאנית המקבלת משתתף מטיפוס **Participant** ומחזירה true אם ההוספה הצליחה, או false אחרת.

(2 נקודות)

הטיפוס **Winner** מאופיין על-ידי שם המשתתף- name, והסכום שעליו לשלם בהתאם לכללי מכרז ויקרי lastPrice.

כותרת הפעולה הבונה במחלקה Winner:

Java:

```
public Winner(String name, int lastPrice)
```

C#:

```
public Winner(string name, int lastPrice)
```

ג. כתבו פעולה פנימית במחלקה **VickeryAuction** המחזירה עצם מטיפוס Winner.

שימו לב! הזוכה הוא זה שהציע את ההצעה הגבוהה ביותר אך סכום התשלום שעליו לשלם הוא של ההצעה השנייה בגובהה.

ניתן להניח שבכל מכרז השתתפו לפחות שני משתתפים ויש רק זוכה אחד.

(7 נקודות)

9. מחרוזת "יפה" היא מחרוזת אשר מורכבת מאותיות גדולות וקטנות בלבד כך שכמות אותיות a שווה לכמות אותיות A, כמות אותיות b שווה לכמות אותיות B וכו'.

לדוגמה: מחרוזת `faBdBAbbDF` היא מחרוזת "יפה".

כתבו פעולה אשר מקבלת כפרמטר מחרוזת ובודקת אם היא "מחרוזת יפה". אם כן - הפעולה תחזיר ערך `true` ואם לא, הפעולה תחזיר `false`.

תזכורת: באלפבית האנגלי יש 26 אותיות גדולות ו- 26 אותיות קטנות.

10. שני מספרים שלמים וחיוביים נקראים "זוג זהב" אם ספרות הקצה שלהם שוות והם גם בעלי מספר זהה של ספרות.

לדוגמה: מספרים 17723 ו- 18113, 3456 ו- 3786, 34 ו- 34 מהווים "זוגות זהב".

א. כתבו פעולה המקבלת שני מספרים שלמים חיוביים ומחזירה `true` אם הם מהווים "זוג זהב" ואם לא, הפעולה תחזיר `false`.

(5 נקודות)

ב. כתבו פעולה המקבלת מערך חד-ממדי ומדפיסה את כל ה"זוגות זהב" הקיימים במערך.

אם אין אף "זוג זהב" תודפס הודעה מתאימה.

(5 נקודות)

ג. מהן סיבוכיות של הפעולות שכתבתם בסעיפים א' ו- ב'? בתשובה יש להתייחס גם לגודל המערך וגם למספר הספרות באיברי המערך.

(2 נקודות)

11. נגדיר "מספר מיוחד" כמספר **חיובי** בעל **שתי ספרות לפחות** שסכום הספרות שלו **במקומות** האי זוגיים שווה לסכום הספרות שלו **במקומות** הזוגיים.

א. כתבו פעולה סטטית שחתימתה:

Java:

```
public static boolean specialNumber(int num)
```

C#:

```
public static bool SpecialNumber(int num)
```

הפעולה מקבלת מספר שלם num, ומחזירה true אם המספר הוא "מספר מיוחד", ובכל מקרה אחר מחזירה false.

דוגמאות:

- עבור הזימון `specialNumber(12551)` הפעולה מחזירה true ($1 + 5 + 1 = 2 + 5$).
- עבור הזימון `specialNumber(4254)` הפעולה מחזירה false ($4 + 5 \neq 2 + 4$).
- עבור הזימון `specialNumber(4)` הפעולה מחזירה false (כי המספר חד ספרתי).
- עבור הזימון `specialNumber(-561)` הפעולה מחזירה false (כי הוא לא חיובי).

(4 נקודות)

מערך של מספרים שלמים נקרא "מערך מיוחד" אם בתחילתו נמצאים ברצף "מספרים מיוחדים" (אם יש כאלה) ואחריהם מספרים לא מיוחדים (אם יש כאלה).

לדוגמה:

- המערך `arr = {12551, 6215, 77, 4254, 4, -561}` הוא "מערך מיוחד" (המספרים המיוחדים: 12551, 6215, 77).
- המערך `brr = {12551, 4, 6215, 77, 4254, -561}` הוא לא "מערך מיוחד" (המספרים המיוחדים: 12551, 6215, 77).

ב. כתבו פעולה `isSpecialArray(int[] arr)`. הפעולה מקבלת מערך מספרים שלמים ומחזירה true אם הוא "מערך מיוחד", ולא הפעולה תחזיר false.

(7 נקודות)

ג. כתבו פעולה `buildSpecialArray(int[] arr1, int[] arr2)`. הפעולה מקבלת שני מערכים של מספרים שלמים `arr1, arr2` ומחזירה מערך חדש שהוא "מערך מיוחד" הכולל את כל הערכים שנמצאים ב- `arr1` ואת כל הערכים שנמצאים ב- `arr2`.

(7 נקודות)

12. הנהלת נמל התעופה "בגין" החליטה למחשב את נתוני הטיסות הממריאות בכל יום בנמל.

לצורך כך נגדיר שלוש מחלקות: Airport, Flight, Time

המחלקה **Time** מייצגת נקודת זמן לפי מספר שעות ומספר דקות.

Java:

```
public class Time
{
  public Time(int hour, int minute){...}
  public boolean before(Time other) {...}
  ...
}
```

C#:

```
public class Time
{
  public Time(int hour, int minute){...}
  public bool Before(Time other) {...}
  ...
}
```

המחלקה מכילה:

- בנאי המקבל כפרמטרים שני מספרים המייצגים את השעה (בתחום 0-23), ואת הדקה (בתחום 0-59).
- פעולה בוליאנית `before(Time other)` המקבלת הפנייה לאובייקט אחר (other) מסוג Time ומחזירה true אם הזמן המיוצג על-ידי האובייקט שמפעיל את הפעולה (this) **קודם** לזמן המיוצג באמצעות other ולא הפעולה מחזירה false.
- במחלקה Time הוגדרו פעולות get ו- set ופעולה toString.
- המחלקה **Time** כבר נתונה ואפשר להשתמש בפעולות שבה **אין צורך לממש אותן!**

המחלקה Flight מייצגת טיסה.

תכונות המחלקה הן:

- code - קוד בין לאומי של הטיסה, מטיפוס String.
- destination - שם העיר שבה נוחתת הטיסה, מטיפוס String.
- departureTime - זמן ההמראה של הטיסה, מטיפוס Time.
- duration - משך זמן הטיסה בדקות, מטיפוס int.
- int numOfPassengers - מספר הנוסעים בטיסה, מטיפוס int.
- isFull - האם הטיסה מלאה, מטיפוס boolean.

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

כמו כן קיים במחלקה קבוע שלם MAX_CAPACITY המציין את המספר המקסימלי של נוסעים על טיסה.

במחלקה Flight הוגדרו פעולות get ו- set ופעולה toString. **אין צורך לממש אותן!**

א. כתבו למחלקה Flight את הפעולות הבאות:

(1) פעולה הבונה (בנאי, constructor) שמקבל כפרמטרים:

קוד טיסה, שם עיר הנחיתה, שני מספרים שלמים המהווים את שעת ההמראה של הטיסה, מספר שלם המייצג את משך זמן הטיסה בדקות ומספר שלם המייצג את מספר הנוסעים בטיסה.

הערך של התכונה הבוליאנית נקבע לפי מספר הנוסעים והקבוע המציין את מספר הנוסעים המקסימלי.

אפשר להניח שכל הפרמטרים תקינים ואין צורך לבדוק זאת.

(2) פעולה בוליאנית המקבלת מספר של נוסעים num, ומוסיפה אותם לטיסה, אם יש בה מקום ומחזירה true. אם אין מקום לכולם - הפעולה לא מוסיפה אף אחד, ומחזירה false.

שימו לב שצריך לעדכן גם את התכונה הבוליאנית isFull במקרה שנדרש לשנותה.

Java:

```
public boolean addPassengers (int num)
```

C#:

```
public bool AddPassengers (int num)
```

(3) פעולה המחשבת **ומחזירה** את זמן הנחיתה של הטיסה.

Java:

```
public Time getArrivalTime()
```

C#:

```
public Time GetArrivalTime()
```

(6 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

המחלקה Airport מייצגת את לוח הטיסות בשדה התעופה ביממה.

הייצוג נעשה על-ידי מערך ששומר את רשימת הטיסות. התכונות במחלקה הן:

- מערך של הטיסות: `Flight [] flightsSchedule`
- מספר הטיסות בלוח הטיסות (במערך): `int numOfFlights`

הטיסות (כלומר האובייקטים מהמחלקה Flight) נמצאים במערך ברצף, **ללא "חורים" מתחילת המערך.**

המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה.

ב. כתבו במחלקה Airport את הפעולות הבאות:

(1) פעולה `firstFlightToPlace` המקבלת עיר כלשהי place, מחזירה את הזמן בו ממריאה הטיסה הראשונה למקום place. אם אין אף טיסה למקום place יוחזר null.

כותרת הפעולה:

Java:

```
public Time firstFlightToPlace(String place)
```

C#:

```
public Time FirstFlightToPlace(string place)
```

(2) פעולה `movePassengers` מקבלת קוד טיסה codeF ומספר שלם num המייצג מספר נוסעים שרוצים להקדים את הטיסה שלהם. הקדמת הטיסה אפשרית אם מתקיימים תנאים הבאים:

- קיימת טיסה אחרת לאותה עיר הנחיתה וטיסה זו **נוחתת לפני** הטיסה codeF.
- יש מקום פנוי לכל num נוסעים הרוצים להקדים את הטיסה. אי אפשר להעביר רק חלק מהנוסעים.

אם אפשר להעביר את כל num נוסעים לטיסה אחרת, הפעולה תדפיס קוד של הטיסה החדשה שלהם ותחזיר true ואם לא, הפעולה תדפיס הודעה מתאימה ותחזיר false.

כותרת הפעולה:

Java:

```
public boolean movePassengers(String codeF, int num)
```

C#:

```
public bool MovePassengers(string codeF, int num)
```

שימו לב, הפעולה צריכה לבצע את כל עדכוני הנתונים הנדרשים.

(12 נקודות)

13. נתונות שלוש הפעולות הבאות:

Java:

```
public static char fun1(String s){
  while (s.length() > 1){
    if (s.charAt(1) < s.charAt(0))
      s = s.substring(1);
    else
      s = s.charAt(0) + s.substring(2);
  }
  return s.charAt(0);
}
public static String fun2(String s, char c){
  int i=0;
  while( i < s.length() && s.charAt(i) != c){
    i++;
  }
  if (i < s.length())
    s = s.substring(0, i) + s.substring(i + 1);
  return s;
}
public static String something(String s){
  String ans = "";
  while (s.length() > 0){
    char a = fun1(s);
    s = fun2(s, a);
    ans = ans + a;
  }
  return ans;
}
```

C#:

```
public static char Fun1(string s){
  while (s.Length > 1){
    if (s[1] < s[0])
      s = s.Substring(1);
    else
      s = s[0] + s.Substring(2);
  }
  return s[0];
}
public static string Fun2(string s, char c){
  int i=0;
```

```
  while( i < s.Length && s[i] != c){
    i++;
  }
  if (i < s.Length)
    s = s.Substring(0, i) + s.Substring(i + 1);
  return s;
}
public static string Something(string s){
  string ans = "";
  while (s.Length > 0){
    char a = Fun1(s);
    s = Fun2(s, a);
    ans = ans + a;
  }
  return ans;
}
```

א. עקבו בעזרת טבלת מעקב אחרי זימון הפעולה `fun1("BANANA")` ורשמו מה תחזיר הפעולה.

(5 נקודות)

ב. עקבו בעזרת טבלת מעקב אחרי זימון הפעולה `fun2('N',"BANANA")` ורשמו מה תחזיר הפעולה.

(5 נקודות)

ג. עקבו בעזרת טבלת מעקב אחרי זימון הפעולה `something("BANANA")` ורשמו מה תחזיר הפעולה. אין צורך לעקוב אחרי פעולות fun1 ו- fun2.

(5 נקודות)

ד. מה מבצעת הפעולה something באופן כללי?

(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

14. הגדרה:

- מסגרת מערך הדו-ממדי מורכבת מתאים הנמצאים בשורה 0, עמודה 0, שורה אחרונה ועמודה אחרונה.

- "תא פנימי" במערך דו-ממדי הוא איבר שלא נמצא במסגרת המערך.

- **תא פנימי** במערך נקרא "עצוב" אם אין אף שכן (מתוך שמונת השכנים) שערכו זהה לערך התא. ערך שנימצא ב"תא עצוב" נקרא "ערך עצוב".

לדוגמא:

- תא (1,1) הוא "תא עצוב" כי ערכים של כל שכנים שלו שונים מערכו. שכנים של תא (1,1) מסומנים באפור. ערך 10 הוא "ערך עצוב".
- גם תא (3,2) הוא תא "עצוב", ערך 21 הוא ערך עצוב.
- גם תא (1,4) הוא תא "עצוב", ערך 123 הוא ערך עצוב.
- תא (2,3) או תא (3,3) הם לא "עצובים".

	0	1	2	3	4	5
0	5	4	1	0	0	3
1	1	10	7	1	123	3
2	6	1	7	0	0	2
3	1	10	21	5	5	2
4	1	10	1	10	10	1

א. כתבו פעולה אשר מקבלת מערך דו-מימדי arr ומספר שלם num. הפעולה תחזיר true אם קיים במערך arr "תא עצוב" שהערך שלו num, ואם לא הפעולה תחזיר false.

(6 נקודות)

ב. כתבו פעולה אשר מקבלת מערך דו-ממדי arr ומחזירה מערך חד-ממדי הכולל את כל ערכים ה"עצובים". שימו לב שבמערך החדש כל ערך צריך להופיע פעם אחת בלבד. אם במערך arr אין תאים "עצובים", יש להחזיר null.

(6 נקודות)

ג. כתבו פעולה אשר מקבלת מערך דו-ממדי של מספרים שלמים. הפעולה תדפיס את **מספר הדו-ספרתי החיובי** הגדול ביותר שעבורו קיים במערך "תא עצוב". אם אין במערך "תא עצוב" שערכו הוא מספר דו-ספרתי חיובי, תודפס הודעה מילולית מתאימה.

עבור דוגמה הנ"ל הפעולה תדפיס 21.

(6 נקודות)

כל מבחני מה"ט עם פתרונות מלאים: www.guyai.co.il/mahat