



תאריך: 17.6.2026

כיתה יא-1

שם התלמיד/ה: _____

שם המורה: צבי מלמד

קרא/י בעיון את השאלות שלפניך, וענה/י עליהן כנדרש. שים/י לב לכל חלקי השאלה. **שמור/י על כללי ההתנהגות הנאותים** בשעת מבחן בעיקרם: **שקט ויושר אישי.**

בחינת מתכונת יסודות במדעי המחשב כיתה יא'-1 – מועד ב'
משך הבחינה (ללא תוספת זמן) 2:30 שעות

- בבחינה נתונות 6 שאלות.
- משקל כל השאלות זהה - 25 נקודות.
- עליכם לענות על 4 שאלות מתוכן.

הנחיות

- ↔ כל חומר עזר, למעט מחשב, מותר לשימוש
- ↔ אין בשום מקרה אפשרות להעביר חומר עזר בין התלמידים
- ↔ יש לענות על הבחינה בעת כחול או שחור או בעיפרון, לעונים בעיפרון לא תינתן זכות הערעור
- ↔ ענו על כל שאלה בדף נפרד. רשמו בראש הדף את מספר השאלה. אם זה המשך התשובה – סמנו בהתאם, כך שיהיה ברור לבודק.
- ↔ **תשובה בכתב יד לא ברור, או מרובת קשקושים ומחיקות – לא תיבדק ותקבל אפס נקודות !!**
- ↔ **יש להקפיד על כללי כתיבה נאותים (לדוגמא הזחה) – הקוד צריך להיות קריא וברור.**
- ↔ **במידת הצורך – העתיקו את התשובה לדף חדש ורישמו באופן ברור X על דף תשובה שלא צריך להיבדק.**

הערות כלליות תכנותיות:

- ↔ הקפידו על ההוראות ושימו לב לניסוחים של השאלות.
- ↔ סמנו V בטבלה להלן – לאיזה שאלות עניתם.
- ↔

| שאלה | סימון – שאלה שבחרתי לענות | ציון | הערות / משוב |
|------|---------------------------|------|--------------|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| סה"כ | | | |

שאלה #1 (25 נק')

נתון קטע הקוד הבא:

```
public static bool Sod1(int x, int y, int z, int k)
{
    int val = (x + y) / 2;

    if (Math.Abs(val - z) > k)
        return true;
    else
        return false;
}

public static int Sod2(int[] arr, int k)
{
    int count = 0;
    bool flag;

    for (int i = 1; i < arr.Length - 1; i++)
    {
        flag = Sod1(arr[i - 1], arr[i + 1], arr[i], k);
        if (flag)
        {
            arr[i] = (arr[i - 1] + arr[i + 1]) / 2;
            count++;
        }
    }

    return count;
}
```

סעיף א

מריצים את קטע הקוד הבא:

```
int[] arr = { 10, 12, 40, 16, 18, 20, 22 };
int num = Sod2(arr, 5);
```

עקבו בעזרת טבלת מעקב. אחרי הקוד שמתבצע בפעולה Sod2

עליכם לוודא שאתם עוקבים בטבלה אחר כל הערכים והתנאים כפי שמקובל וכמו כן וודאו שאתם עוקבים אחרי הערכים של $arr[i-1]$, $arr[i]$, $arr[i+1]$.

הערה: אין צורך לעקוב אחר ביצוע הפעולה Sod1

סעיף ב'

רשמו את תוכן המערך בסיום ריצת הקוד ואת הערך של המשתנה num

סעיף ג'

כתבו במשפט אחד מה הפעולה Sod2 מבצעת. הקפידו על ניסוח ברור.

סעיף ד'

הגדירו מערך בגודל 7, כך שהקריאה לפעולה תחזיר את הערך 3.

שאלה #2 (25 נק')

נגדיר **רצף עולה-חזק** כרצף של לפחות 3 איברים סמוכים במערך, המקיים:

1. כל איבר גדול מהאיבר שלפניו.
2. ההפרש בין כל שני איברים סמוכים ברצף גדול או שווה להפרש שקדם לו.

דוגמאות:

- א. המערך {2, 5, 9, 14, 21} ← כולו רצף עולה-חזק.
- ב. המערך {1, 4, 8, 7, 10, 14} ← מכיל שני רצפים עולים-חזקים: {1, 4, 8} ו- {7, 10, 14}.
- ג. המערך {3, 6, 10, 15, 16, 20} ← מכיל רצף עולה-חזק אחד באורך 4: {3, 6, 10, 15}.
- ד. המערך {2, 6, 9, 11} ← לא מכיל רצף עולה-חזק, כי ההפרשים הם 4, 3, 2.

סעיף א'

כתבו פעולה סטטית בוליאנית בשם `IsStrongAscend` המקבלת מערך של מספרים שלמים – `arr`. הפעולה תחזיר `true` אם כל המערך הוא רצף עולה חזק, אחרת תחזיר `false`.

סעיף ב'

כתבו פעולה סטטית בשם `LongestStrongAscend` המקבלת מערך של מספרים שלמים – `arr`. הפעולה תחזיר את האורך של הרצף המתחלף הארוך ביותר במערך.

דוגמה: עבור המערכים בדוגמאות הנ"ל: בדוגמא א' הפעולה תחזיר 5, בדוגמא ב' הפעולה תחזיר 3, בדוגמא ג' היא תחזיר 4 ובדוגמא ד' היא תחזיר 0.

שאלה #3 (25 נק')

סעיף א

כתוב פעולה חיזונית `int DigitK(int num, int k)`. הפעולה מחזירה את הספרה ה-`k`-ית של המספר. אם במספר אין `k` ספרות אז הפעולה תחזיר `-1`.

לדוגמא:

`DigitK(5678,1)` ← מחזירה 8

`DigitK(5678,4)` ← מחזירה 5

`DigitK(5678,0)` ← מחזירה -1

`DigitK(5678,5)` ← מחזירה -1

סעיף ב

שני מספרים חיוביים נקראים **מספרי מראה** אם הם בעלי אותו מספר ספרות, והספרות של אחד מהם מופיעות בסדר הפוך במספר השני.

דוגמאות:

- 1234 ו-4321 הם מספרי מראה.
- 785 ו-587 הם מספרי מראה.
- 1221 ו-1221 הם מספרי מראה.
- 120 ו-21 אינם מספרי מראה (מספר ספרות שונה).
- 123 ו-321 אינם מספרי מראה (מספר ספרות שונה).

כתבו את הפעולה `public static bool IsMirror(int num1, int num2)`. הפעולה תקבל שני מספרים חיוביים ותחזיר `true` אם הם מספרי מראה זה של זה, אחרת תחזיר `false`.

הערה: ניתן להיעזר בפעולה שכתבתם בסעיף א', אך יתקבל גם פתרון נכון אחר.

סעיף ג'

כתוב את הפעולה `public static int CountMirrorPairs(int[] arr)`. הפעולה תקבל מערך של מספרים חיוביים ותחזיר את מספר זוגות מספרי המראה במערך.

שני איברים במערך מהווים זוג אם הם נמצאים **במקומות שונים במערך** והם מספרי מראה זה של זה.

לדוגמה, עבור המערך: { 111, 111, 54, 321, 45, 123 }

הפעולה תחזיר: 3, מפני שקיימים שלושה זוגות מספרי מראה: 123 ו-321, 45 ו-54, 111 ו-111

שאלה #4 (25 נק')

מועדון הטיסה "טסים בכיף" מעניק הטבות שונות לחברי המועדון בהתאם למספר הנקודות שצברו ולוותק שלהם במועדון.

נתונה המחלקה FlightMember שלה חמש תכונות:

- name – שם חבר המועדון, מטיפוס מחרוזת.
- points – מספר הנקודות שצבר חבר המועדון, מטיפוס שלם
- months – מספר החודשים שבהם חבר המועדון רשום במועדון, מטיפוס שלם.
- flights – מספר הטיסות שביצע חבר המועדון, מטיפוס שלם.
- active – ערך בוליאני המציין האם החבר פעיל במועדון.

הניחו שקיימות פעולות Get/Set לכל תכונות המחלקה.

סעיף א

כתבו פעולה פנימית במחלקה FlightMember בשם IsEligible המקבלת מספר שלם neededPoints הפעולה תחזיר true אם חבר המועדון זכאי לכרטיס טיסה חינם, או false אחרת.

חבר מועדון זכאי לכרטיס חינם אם מתקיימים כל התנאים הבאים:

- חבר המועדון פעיל במועדון.
- חבר המועדון רשום לפחות 6 חודשים.
- מספר הנקודות שלו גדול או שווה ל-neededPoints.
- חבר המועדון ביצע לפחות 3 טיסות.

סעיף ב

נתונה המחלקה FlightClub שלה התכונה הבאה:

- members - מערך מטיפוס FlightMember

הניחו כי המערך ממוין לפי סדר הכניסה למועדון, ללא ערכי null.

הניחו שקיימות פעולות Get/Set לכל תכונות המחלקה.

במחלקה FlightClub כתבו פעולה בשם CountEligible המקבלת מספר שלם neededPoints . הפעולה תחזיר כמה חברי מועדון זכאים לכרטיס טיסה חינם.

סעיף ג

כתבו פעולה חיצונית בשם BestMember המקבלת מערך members מטיפוס FlightMember

הפעולה תחזיר את שמו של חבר המועדון שצבר את מספר הנקודות הגבוה ביותר.

אם קיימים כמה חברים בעלי אותו מספר נקודות מקסימלי, יש להחזיר את שמו של הראשון מביניהם במערך.

שאלה #5 (25 נק')

נתונה המחלקה Time להלן:

```
Class Time {  
    private int hour, min, sec; // שעות, דקות, שניות  
}
```

והוגדרו פונקציות Get/Set המתאימות (אין צורך לכתוב אותן), וכמו כן אין צורך לכתוב את הבנאים פרט למה שנדרש במפורש בהמשך.

סעיף א'

1. הוסיפו למחלקה פעולה `GetTotalSeconds()` שמחזירה את מספר השניות הכולל מאז חצות (כלומר מאז השעה `00:00:00`).

2. הוסיפו למחלקה בנאי `public Time(Time other, int secondsToAdd)` שיוצר יוצר אובייקט זמן חדש שמתקבל לאחר הוספת מספר השניות הנתון. ניתן להניח שהזמן החדש אינו עובר את חצות.

סעיף ב'

נתונה המחלקה `RunRace`, תחרות-ריצה או בקיצור מרוץ. המחלקה מכילה את שם המתחרה מטיפוס מחרוזת, התכונה `startTime` מטיפוס `Time`, משתנה בוליאני `ended` שמציין האם המרוץ כבר הסתיים, ומשתנה `duration` מטיפוס `int` שמתאר כמה זמן נמשכה הריצה. כאשר נוצר אובייקט חדש המשתנה `ended` מכיל את הערך `false` והמשתנה `duration` מכיל את הערך `אפס`. כאשר הריצה מסתיימת המשתנה `ended` מקבל ערך `true` והמשתנה `duration` מקבל את הערך הדרוש. נתונים הבנאים והפונקציות `Get/Set` ואין צורך לממש אותן.

1. כתבו פעולה פנימית בשם `void EndRace(Time endTime)`. הפעולה מעדכנת את מצב המרוץ לכך שהמירוץ הסתיים ואת משך הזמן.

2. כתבו פעולה פנימית: `public Time GetEndTime()`. הפעולה מחזירה אובייקט מטיפוס `Time` המייצג את זמן סיום המרוץ. אם המרוץ עדיין לא הסתיים היא תחזיר `null`.

סעיף ג'

כתבו פעולה חיצונית: `public static string GetWinner(RunRace[] arr)`. הפעולה מחזירה את שם המתחרה שהשיג את התוצאה הטובה ביותר (כלומר, משך הריצה הקטן ביותר מבין המרוצים שהסתיימו). הערה: ניתן להניח שהמערך אינו מכיל ערכי `null`.

שאלה #6 (25 נק')

נתונה המחלקה Msg המייצגת הודעה. לכל הודעה יש:

- תוכן ההודעה (מחרוזת).
- מצב המציין האם ההודעה נקראה.

המחלקה כוללת את הפעולות הבאות:

```
public Msg(string text)
```

בנאי היוצר הודעה חדשה. בעת יצירת ההודעה היא מסומנת כהודעה שטרם נקראה.

```
public string GetText()
```

מחזירה את תוכן ההודעה.

```
public void ReadMsg()
```

מסמנת את ההודעה כהודעה שנקראה.

```
public bool IsRead()
```

מחזירה true אם ההודעה נקראה, אחרת מחזירה false.

```
public override string ToString()
```

מחזירה מחרוזת המייצגת את ההודעה.

אין צורך לממש את המחלקה Msg.

נדרש לממש מחלקה בשם Messages המייצגת מאגר הודעות.

למחלקה התכונות הבאות:

```
private Msg[] messages;
```

```
private int count;
```

- messages - מערך של הודעות מסוג Msg.
- count - מספר ההודעות השמורות כרגע במאגר.

הניחו:

- ההודעות נשמרות במערך מהאינדקס 0 ועד 1 - count.
- יתר תאי המערך אינם בשימוש.
- אין צורך לכתוב פעולות Get/Set.

כל הסעיפים הבאים מתייחסים לפעולות פנימיות של המחלקה Messages.

סעיף א'

כתבו בנאי למחלקה שמקבל את מספר ההודעות המרבי שניתן לשמור במאגר ויאתחל את התכונות הנדרשות.

סעיף ב'

כתבו פעולה פנימית `public bool AddMsg(string text)`. אם יש מקום פנוי במאגר אז הפעולה תיצור אובייקט חדש מסוג `Msg`, תוסיף אותו למאגר ותחזיר `true`. אחרת, הפעולה תחזיר `false` מבלי לשנות את המאגר.

סעיף ג'

כתבו פעולה: `public Msg ReadNextUnreadMsg()`

הפעולה תחפש את ההודעה הראשונה במאגר שטרם נקראה. אם נמצאה הודעה כזו אז הפעולה תסמן אותה כהודעה שנקראה ותחזיר אותה. אם כל ההודעות במאגר כבר נקראו אז הפעולה תחזיר `null`.

סעיף ד'

כתבו פעולה פנימית: `public void ClearAllReadMessages()`. הפעולה תמחק מן המאגר את כל ההודעות שנקראו. לאחר סיום הפעולה במאגר יישארו רק הודעות שטרם נקראו.

