

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.
פרק ראשון – בפרק זה ארבע שאלות, – (25×2) – 50 נקודות
ומהן יש לענות על שתיים.
פרק שני – בפרק זה שאלות בארבעה מסלולים שונים – (25×2) – 50 נקודות
ענה על שאלות רק במסלול שלמדת,
לפי ההוראות בקבוצת השאלות במסלול זה.
סה"כ – 100 נקודות
- ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.
- ד. הוראות מיוחדות:
- את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון
כתוב בשפה אחת בלבד – C# או Java.
 - רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – C# או Java.
 - רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת,
אחד מארבעת המסלולים: מערכות מחשב ואסמבלר, מבוא לחקר ביצועים,
מודלים חישוביים, תכנות מונחה עצמים.
- הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום
אות קטנה או להפך.
- כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטיטה (ראשי פרקים, חישובים וכדומה).
רשום "טיטה" בראש כל עמוד טיטה. רישום טיטות כלשהן על דפים שמוחץ למחברת הבחינה עלול לגרום לפסילת הבחינה!
- ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.**

בהצלחה!

השאלות

בשאלון זה שני פרקים: פרק ראשון ופרק שני.
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

פרק ראשון (50 נקודות)

שיום לב: בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות `BinTreeNode<T>`, `Stack<T>`, `Queue<T>`, `Node<T>`, בלי לממש אותן. אם אתה משתמש בפעולות נוספות עליך לממש אותן.

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

שיום לב: לשאלה זו שני נוסחים: אחד ב-Java (עמודים 2-3), ואחד ב-C# (עמודים 4-5). עבוד על פי השפה שלמדת.

1.

לפותרים ב-Java

א. לפניך כמה מחלקות. עקוב אחר הפעולה main שבמחלקה Program ורשום את הפלט. במעקב יש לכלול את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```
public class One
{
    private double n;
    public One(double n) { this.n = n; }
    public double getN() { return this.n; }
    public void f() { System.out.println("f of one "); }
    public void g() { System.out.println("g of one " + this.getN()); }
}
```

```
public class Two
{
    private Queue<One> q;
    public Two(int n , int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.insert(new One(Math.pow(2 , i)));
    }
    public void f()
    {
        System.out.print("f of two ");
        if (!this.q.isEmpty())
            System.out.println(this.q.remove().getN());
    }
}
```

/המשך בעמוד 3/

```

public class Together
{
    private double x;
    private int from , to;
    private One first;
    private Two second;

    public Together(double x , int from , int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from , to);
    }

    public void methodA()
    {
        System.out.println("-- MethodA() -- ");
        this.first.f();
    }
    public void methodB()
    {
        System.out.println("-- MethodB() -- ");
        this.first.g();
        this.second.f();
    }
}

public class Program
{
    public static void main(String[] args)
    {
        Together tg = new Together(5.0 , 2 , 6);
        tg.methodA() ;
        tg.methodB() ;
    }
}

```

- ב. בהנחה כי הפעולה הבונה במחלקה Together מקבלת מספרים גדולים מ-0 בלבד –
 מה מבצעת הפעולה f() במחלקה Two ?
 /המשך בעמוד 4/

א. לפניך כמה מחלקות. עקוב אחר הפעולה Main שבמחלקה Program ורשום את הפלט. במעקב יש לכלול את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```
public class One
{
    private double n;
    public One(double n) { this.n = n; }
    public double GetN() { return this.n;}
    public void F()      { Console.WriteLine("F of one "); }
    public void G()      { Console.WriteLine("G of one " + this.GetN()); }
}
```

```
public class Two
{
    private Queue<One> q;
    public Two(int n , int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.Insert(new One(Math.Pow(2 , i)));
    }

    public void F()
    {
        Console.Write("F of two ");
        if (!this.q.IsEmpty())
            Console.WriteLine(this.q.Remove().GetN());
    }
}
```

/המשך בעמוד 5/

```

public class Together
{
    private double x;
    private int from , to;
    private One first;
    private Two second;

    public Together(double x , int from , int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from , to);
    }

    public void MethodA()
    {
        Console.WriteLine("-- MethodA() -- ");
        this.first.F();
    }

    public void MethodB()
    {
        Console.WriteLine("-- MethodB() -- ");
        this.first.G();
        this.second.F();
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Together tg = new Together(5.0 , 2 , 6);
        tg.MethodA();
        tg.MethodB();
    }
}

```

2. בהנחה כי הפעולה הבונה במחלקה Together מקבלת מספרים גדולים מ-0 בלבד –

מה מבצעת הפעולה F() במחלקה Two ?
/המשך בעמוד 6/

שיום לב: לשאלה זו שני נוסחים: אחד ב-Java (עמודים 6-7), ואחד ב-C# (עמודים 8-9). עבוד על פי השפה שלמדת.

.2

לפותרים ב-Java

לפניך הפעולות sod ו-what המקבלות מערך a שאיבריו מטיפוס שלם, ממוין בסדר עולה, ומספר שלם k. לשתי הפעולות אותה טענת יציאה.

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length-1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

נתון מערך a :

2	4	7	12	18
---	---	---	----	----

- א.** עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה sod בעבור המערך הנתון a והמספר $k = 11$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול עמודות בעבור: $a[i]$, $a[j]$, i , j , ועמודה נוספת שבה יצוין אם התנאי שבפקודת if מתקיים או אינו מתקיים.
- ב.** עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה sod בעבור המערך הנתון a והמספר $k = 10$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול את העמודות שפורטו בסעיף א.
- ג.** מהי טענת היציאה של הפעולה sod ?
- ד.** מהי סיבוכיות זמן הריצה של הפעולה sod ? נמק את תשובתך.
- ה.** עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה what בעבור המערך הנתון a והמספר $k = 11$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול עמודות בעבור: $a[\text{left}]$, $a[\text{right}]$, left, right, ושתי עמודות נוספות לכל אחת מפקודות if. בכל עמודה יצוין אם התנאי בפקודת if מתקיים או אינו מתקיים.
- ו.** מהי סיבוכיות זמן הריצה של הפעולה what ? נמק את תשובתך.
- ז.** מי מבין שתי הפעולות sod או what – יעילה יותר? נמק את תשובתך.
- ח.** טענת הכניסה של הפעולות sod ו-what שונתה כך שאפשר להעביר אליהן מערך a לא ממורן.
- (1)** האם טענת היציאה של הפעולה sod תשתנה? נמק את תשובתך.
- (2)** האם טענת היציאה של הפעולה what תשתנה? נמק את תשובתך.

לפותרים ב- C#

לפניך הפעולות Sod ו- What המקבלות מערך a שאיבריו מטיפוס שלם, ממיון בסדר עולה, ומספר שלם k. לשתי הפעולות אותה טענת יציאה.

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length-1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```


נתון מערך a :

2	4	7	12	18
---	---	---	----	----

- א.** עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון a והמספר $k = 11$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול עמודות בעבור: $i, j, a[i], a[j]$, ועמודה נוספת שבה יצוין אם התנאי שבפקודת if מתקיים או אינו מתקיים.
- ב.** עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון a והמספר $k = 10$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול את העמודות שפורטו בסעיף א.
- ג.** מהי טענת היציאה של הפעולה Sod ?
- ד.** מהי סיבוכיות זמן הריצה של הפעולה Sod ? נמק את תשובתך.
- ה.** עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה What בעבור המערך הנתון a והמספר $k = 11$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול עמודות בעבור: $left, right, a[left], a[right]$, ושתי עמודות נוספות לכל אחת מפקודות if. בכל עמודה יצוין אם התנאי בפקודת if מתקיים או אינו מתקיים.
- ו.** מהי סיבוכיות זמן הריצה של הפעולה What ? נמק את תשובתך.
- ז.** מי מבין שתי הפעולות – Sod או What – יעילה יותר? נמק את תשובתך.
- ח.** טענת הכניסה של הפעולות Sod ו-What שונתה כך שאפשר להעביר אליהן מערך a לא ממוין.
- (1)** האם טענת היציאה של הפעולה Sod תשתנה? נמק את תשובתך.
- (2)** האם טענת היציאה של הפעולה What תשתנה? נמק את תשובתך.

3.

שים לב: לשאלה זו שני נוסחים: אחד ב-Java (עמודים 10-11), ואחד ב-C# (עמודים 12-13). עבוד על פי השפה שלמדת.

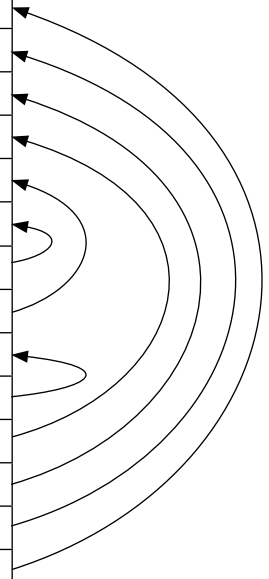
לפתורים ב-Java

לפניך ממשק של המחלקה "תור-ביטול" (UndoQueue):

המחלקה מגדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה והוצאה של ערכים שלמים וגדולים מ-0.

שם הפעולה	תיאור
UndoQueue()	הפעולה בונה תור-ביטול ריק.
boolean isEmpty()	הפעולה מחזירה true אם התור-ביטול הנוכחי ריק, אחרת – הפעולה מחזירה false.
void insert(int x)	הפעולה מכניסה את הערך x לסוף התור-ביטול הנוכחי.
int remove()	הפעולה מוציאה את הערך שבראש התור-ביטול הנוכחי, ומחזירה אותו. <u>הנחה:</u> התור-ביטול הנוכחי אינו ריק.
int head()	הפעולה מחזירה את הערך של האיבר שבראש התור-ביטול בלי להוציאו. <u>הנחה:</u> התור-ביטול הנוכחי אינו ריק.
void undo()	הפעולה מבטלת את פעולת ה- remove או את פעולת ה- insert האחרונה שבוצעה, ומשחזרת את התור-ביטול כפי שהיה לפני ביצוע הפעולה. אם אין פעולות remove או insert שבוצעו, הפעולה אינה עושה דבר. <u>שים לב:</u> הפעלה n פעמים של הפעולה undo, תחזיר את התור-ביטול למצב שבו הוא היה לפני ביצוע n פעולות remove או insert האחרונות שטרם בוטלו.

הפעולה	מצב התור-ביטול לאחר ביצוע הפעולה
UndoQueue uq = new UndoQueue();	התור-ביטול ריק
uq.insert(2);	→2→
uq.insert(3);	→3, 2→
uq.remove();	→3→
uq.insert(4);	→4, 3→
uq.remove();	→4→
uq.remove();	התור-ביטול ריק
uq.undo();	→4→
uq.undo();	→4, 3→
uq.insert(1);	→1, 4, 3→
uq.undo();	→4, 3→
uq.undo();	→3→
uq.undo();	→3, 2→
uq.undo();	→2→
uq.undo();	התור-ביטול ריק
uq.undo();	התור-ביטול ריק



א. לפניך קטע קוד.

```
UndoQueue q = new UndoQueue();
q.insert(1);
q.insert(2);
q.insert(3);
q.remove();
q.insert(4);
q.undo();
q.undo();
```

הראה את מצב התור-ביטול לאחר הביצוע של כל הוראה בקטע הקוד.

ב. כתוב ב-Java את כותרת המחלקה **UndoQueue**, ואת התכונות שלה. רשום תיעוד לכל תכונה.

ג. ממש ב-Java את הפעולות insert, remove ו-undo המוצגות בממשק המחלקה **UndoQueue**.

אתה יכול להשתמש בפעולות הממשק האחרות של **UndoQueue** בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

לפותרים ב- C#

לפניך ממשק של המחלקה "תור-ביטול" (UndoQueue):
 המחלקה מגדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה והוצאה של ערכים שלמים
 וגדולים מ-0.

שם הפעולה	תיאור
UndoQueue()	הפעולה בונה תור-ביטול ריק.
bool IsEmpty()	הפעולה מחזירה true אם התור-ביטול הנוכחי ריק, אחרת – הפעולה מחזירה false.
void Insert(int x)	הפעולה מכניסה את הערך x לסוף התור-ביטול הנוכחי.
int Remove()	הפעולה מוציאה את הערך שבראש התור-ביטול הנוכחי, ומחזירה אותו. <u>הנחה:</u> התור-ביטול הנוכחי אינו ריק.
int Head()	הפעולה מחזירה את הערך של האיבר שבראש התור-ביטול בלי להוציאו. <u>הנחה:</u> התור-ביטול הנוכחי אינו ריק.
void Undo()	הפעולה מבטלת את פעולת ה- Remove או את פעולת ה- Insert האחרונה שבוצעה, ומשחזרת את התור-ביטול כפי שהיה לפני ביצוע הפעולה. אם אין פעולות Remove או Insert שבוצעו, הפעולה אינה עושה דבר. <u>שים-לב:</u> הפעלה n פעמים של הפעולה Undo, תחזיר את התור-ביטול למצב שבו הוא היה לפני ביצוע n פעולות Remove או Insert האחרונות שטרם בוטלו.

מצב התור-ביטול לאחר ביצוע הפעולה	הפעולה
התור-ביטול ריק	UndoQueue uq = new UndoQueue();
→2→	uq.Insert(2);
→3 , 2→	uq.Insert(3);
→3→	uq.Remove();
→4 , 3→	uq.Insert(4);
→4→	uq.Remove();
התור-ביטול ריק	uq.Remove();
→4→	uq.Undo();
→4 , 3→	uq.Undo();
→1 , 4 , 3→	uq.Insert(1);
→4 , 3→	uq.Undo();
→3→	uq.Undo();
→3 , 2→	uq.Undo();
→2→	uq.Undo();
התור-ביטול ריק	uq.Undo();
התור-ביטול ריק	uq.Undo();

א. לפניך קטע קוד.

```
UndoQueue q = new UndoQueue();
q.Insert(1);
q.Insert(2);
q.Insert(3);
q.Remove();
q.Insert(4);
q.Undo();
q.Undo();
```

הראה את מצב התור-ביטול לאחר הביצוע של כל הוראה בקטע הקוד.

ב. כתוב ב-C# את כותרת המחלקה UndoQueue, ואת התכונות שלה. רשום תיעוד לכל תכונה.

ג. ממש ב-C# את הפעולות Insert, Remove, ו-Undo המוצגות בממשק המחלקה UndoQueue.

אתה יכול להשתמש בפעולות הממשק האחרות של UndoQueue בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

4. א. לפניך כותרת של פעולה:

ב- Java :

```
public static void leaves(BinTreeNode<Integer> t , Stack<Integer> s)
```

ב- C# :

```
public static void Leaves(BinTreeNode<int> t , Stack<int> s)
```

הפעולה מקבלת עץ בינרי לא ריק t של מספרים שלמים, ומחסנית ריקה s של מספרים שלמים.

הפעולה מכניסה למחסנית את ערכי כל העלים של העץ t , על פי סדר סריקה מימין לשמאל.

ממש ב- Java או ב- C# את הפעולה.

ב. כתוב ב- Java או ב- C# פעולה בוליאנית שתקבל 2 עצים בינריים לא ריקים

של מספרים שלמים, ותחזיר true אם מתקיימים שני התנאים האלה:

– יש להם אותו מספר עלים

– על פי סדר הסריקה מימין לשמאל, ערכי העלים שווים

אחרת – הפעולה תחזיר false .

עליך להשתמש בפעולה שממשת בסעיף א.

/המשך בעמוד 15/

פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלר, עמ' 15-20

מבוא לחקר ביצועים, עמ' 21-31

מודלים חישוביים, עמ' 32-34

תכנות מונחה עצמים ב-Java, עמ' 36-45; תכנות מונחה עצמים ב-C#, עמ' 46-55

ענה רק על שאלות במסלול שלמדת.

מערכות מחשב ואסמבלר

אם למדת מסלול זה, ענה על שתיים מהשאלות 5-8 (לכל שאלה – 25 נקודות).

5. במחסנית מאוחסנים 10 מספרים שלמים עם סימן, בגודל מילה.

בנוסף, במקטע הנתונים מוגדרים משתנה K ושני מערכים, ARR1 ו-ARR2.

כל תאי המערך ARR1 מאותחלים ל-0, והמשתנה K מאותחל ל-0.

ARR1	DW	10	DUP (0)
ARR2	DW	10	DUP (?)
K	DW	0	

יש להעתיק למערך ARR1 את כל המספרים שמאוחסנים במחסנית באופן הזה:

כל המספרים השליליים יאוחסנו בתחילת המערך לפי סדר הוצאתם מהמחסנית. אחרי

המספרים השליליים יאוחסנו כל האפסים, ולאחר מכן יאוחסנו כל המספרים החיוביים

לפי סדר הוצאתם מהמחסנית.

לדוגמה, בעבור המחסנית שלפניך המכילה 6 איברים:

-5
0
10
-2
0
8

ייראה המערך ARR1 כך:

0	1	2	3	4	5
-5	-2	0	0	10	8

לפניך קטע תכנית באסמבלר, המבצע את הנדרש.

מקטע התכנית הושמטו שורות במקומות המסומנים במספרים i-iv.

רשום במחברתך את מספרי השורות המסומנות, וכתוב ליד כל אחד מהמספרים את

שורת הפקודה החסרה, כדי שקטע התכנית יבצע את הנדרש.

		MOV	SI, 0
		MOV	DI, 0
		MOV	CX, 10
	A1:	POP	AX
i		_____	_____
		JLE	A2
ii		_____	_____
		ADD	DI, 2
		JMP	A4
	A2:	CMP	AX, 0
		JNE	A3
		INC	K
		JMP	A4
iii	A3:	_____	_____
		ADD	SI, 2
	A4:	LOOP	A1
		ADD	SI, K
		ADD	SI, K
		MOV	CX, DI
		SHR	CX, 1
		XOR	BX, BX
	A5:	MOV	DX, ARR2[BX]
		MOV	ARR1[SI], DX
iv		_____	_____
		ADD	BX, 2
		LOOP	A5

6. מערך A נקרא **מוכל** במערך B, אם האורך של מערך A אינו עולה על האורך של מערך B, וכל הערכים של איברי מערך A נמצאים במערך B ברצף ובאותו סדר כמו במערך A. בכל מערך מאוחסנים מספרים שלמים עם סימן, השונים זה מזה. לדוגמה:

בעבור המערכים A ו-B שלפניך, המערך A **מוכל** במערך B. מערך A:

0	1	2	3	4	5
3	-5	0	8	5	11

מערך B:

0	1	2	3	4	5	6	7	8	9
15	-7	4	3	-5	0	8	5	11	2

איברי המערך B המסומנים באפור הם הערכים של איברי המערך A.

א. במקטע הנתונים הוגדרו הנתונים כך:

```
ARR_B DB 100 DUP (?)
ARR_A DB 10 DUP (?)
V DB ?
P DB ?
```

כתוב באסמבלר שגרה (פרוצדורה) בשם TEST, שתקבל ערך שמאוחסן במשתנה V. השגרה תבדוק אם הערך שמאוחסן במשתנה V נמצא במערך ARR_B. אם כן – השגרה תאחסן במשתנה P את האינדקס של האיבר המתאים, אחרת – השגרה תאחסן במשתנה P את הערך -1.

ב. כתוב באסמבלר קטע תכנית שיבדוק אם המערך ARR_A **מוכל** במערך ARR_B. אם כן – קטע התכנית יאחסן 1 באוגר BL, אחרת – הוא יאחסן 0 באוגר BL. עליך להשתמש בשגרה TEST שכתבת בסעיף א. הנח שהאורך של המערך ARR_A אינו עולה על האורך של מערך ARR_B.

7. בשאלה זו שני סעיפים א-ב, שאין קשר ביניהם. ענה על שניהם.

א. במקטע הנתונים הוגדר מערך ARR :

ARR DB 6 DUP (?)

יש להפוך את סדר האיברים במערך.

לדוגמה, נתון המערך:

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

לאחר הפיכת סדר האיברים, ייראה המערך כך:

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

לפניך 2 קטעים באסמבלר, ii-i , שמטרתם להפוך את סדר האיברים

במערך ARR .

עקוב בעזרת טבלת מעקב אחר הביצוע של כל אחד מהקטעים ii-i שלפניך, וקבע אם

הוא מבצע את הנדרש או אינו מבצע את הנדרש.

i

```

MOV SI, 3
MOV DI, 2
A1: MOV AL, ARR[DI]
     MOV AH, ARR[SI]
     ADD ARR[DI], AH
     SUB ARR[DI], AL
     MOV ARR[SI], AL
     INC SI
     DEC DI
     JNZ A1
    
```

ii

```

MOV CX, 3
MOV SI, 3
MOV DI, 2
A1: MOV AL, ARR[DI]
     MOV AH, ARR[SI]
     MOV ARR[DI], AH
     MOV ARR[SI], AL
     INC SI
     DEC DI
     LOOP A1
    
```

ב. (אין קשר לסעיף א.)

במקטע הנתונים הוגדרו הנתונים כך:

```
T DB 55, 90, 110, 1
```

שים לב: המספרים הם עשרוניים.

לפניך קטע תכנית באסמבלר, שמטרתו לחשב את סכום האיברים במערך T.

```
MOV CX, 4
XOR AX, AX
LEA BX, T
AGAIN: ADD AL, [BX]
INC BX
LOOP AGAIN
```

קטע התכנית אינו מבצע את הנדרש.

(1) עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית, וכתוב מה יהיה התוכן של AL, ומה יהיה התוכן של דגל הנשא ושל דגל הגלישה בסיום הביצוע של קטע התכנית.

(2) שנה את קטע התכנית כך שיחשב את סכום האיברים במערך הנתון T. העתק למחברתך את קטע התכנית לאחר השינוי.

8. לפניך קטע תכנית באסמבלר:

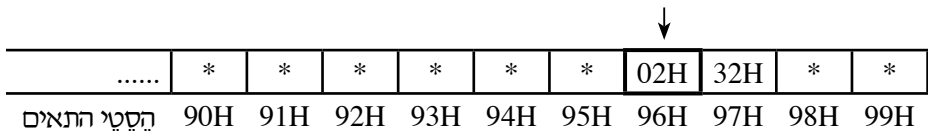
- i PUSHF
- ii PUSH BP
- iii MOV BP, SP
- iv MOV AX, [BP+4]
- v MOV BP, [BP]
- vi ADD SP, 4
- vii POPF

נתון שלפני ביצוע קטע התכנית:

- התוכן של אוגר הדגלים הוא 3202H
- התוכן של האוגר SP הוא 96H
- התוכן של האוגר BP הוא 5678H

לפניך חלק ממחסנית המכילה מספרים הקסדצימליים.

לפני ביצוע קטע התכנית, האוגר SP מצביע על התא המסומן בחץ, שכתובתו היחסית 96H.



תא המסומן ב- * הוא תא שאין משמעות לתוכנו.

א. עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית.

בטבלת המעקב יש לכלול עמודות בעבור האוגרים: AX, BP, SP.

ב. על פי טבלת המעקב, ענה על התת-סעיפים (1)-(4).

(1) איזה מספר יירשם במחסנית הנתונה לאחר ביצוע הוראה i ?

כתוב את הכתובת היחסית של התאים שבהם יירשם המספר.

(2) מה יהיה התוכן של האוגר BP לאחר ביצוע הוראה iii ?

(3) מה יהיה התוכן של האוגר AX לאחר ביצוע הוראה iv ?

(4) מה יהיה התוכן של האוגר BP לאחר ביצוע הוראה v ?

מבוא לחקר ביצועים

אם למדת מסלול זה, ענה על שתיים מהשאלות 9-12 (לכל שאלה – 25 נקודות).

9. בשאלה זו שלושה סעיפים א-ג, שאינם תלויים זה בזה. ענה על כל הסעיפים.

א. יהי $G = (V, E)$ גרף מכוון, V היא קבוצת הקדקודים ו- E היא קבוצת הקשתות.

G מיוצג על ידי מטריצת הסמיכויות שלפניך:

	a	b	c	d	e
a	0	1	1	0	0
b	0	0	1	0	0
c	0	1	0	1	0
d	0	0	0	0	1
e	1	0	1	0	0

(1) סרטט את הגרף G המיוצג על ידי המטריצה.

(2) מצא את רכיב / רכיבי הקשירות החזקה (רק"ח / רק"חים) שבגרף G .

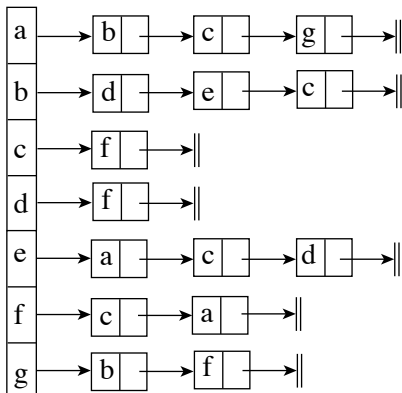
בעבור כל רק"ח שמצאת, רשום את קבוצת הקדקודים שלו.

(3) הסר קשת אחת מהגרף G שסרטטת, כך שמספר הרק"חים יגדל ב-1.

ציין את הקשת שהסרת, ולכל רק"ח רשום את קבוצת הקדקודים שלו.

ב. יהי $G = (V, E)$ גרף מכוון, V היא קבוצת הקדקודים ו- E היא קבוצת הקשתות.

G מיוצג על ידי רשימת הסמיכויות הזו:



(1) הפעל אלגוריתם סריקה לעומק (DFS) על הגרף הנתון החל מקדקוד a.

סרטט במחברתך את העץ הפורש DFS שמתקבל.

(2) הפעל אלגוריתם סריקה לרוחב (BFS) על הגרף הנתון החל מקדקוד d.

סרטט במחברתך את העץ הפורש BFS שמתקבל. / המשך בעמוד 22 /

ג. יהי G גרף מכוון.

נפעיל את האלגוריתם סריקה לעומק (DFS) על הגרף המכוון G .

נגדיר את הקשתות של הגרף G בעזרת העץ הפורש / היער הפורש שנוצר.

קשת עץ היא קשת של הגרף G , הנמצאת גם בעץ הפורש / היער הפורש של G .

קשת אחורית היא קשת (u, v) בגרף G , שאינה נמצאת בעץ הפורש / היער הפורש.

זו קשת מקדקוד u לקדקוד v בגרף G , כאשר v הוא אב קדמון של u

בעץ הפורש / היער הפורש.

קשת קדמית היא קשת (u, v) בגרף G , שאינה נמצאת בעץ הפורש / היער הפורש.

זו קשת מקדקוד u לקדקוד v בגרף G , כאשר v הוא צאצא של u

בעץ הפורש / היער הפורש.

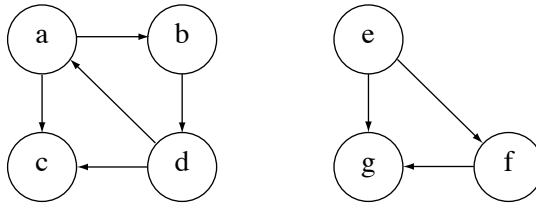
קשת חוצה היא קשת (u, v) בגרף G , שאינה נמצאת בעץ הפורש / היער הפורש,

ואינה **קשת אחורית** או **קשת קדמית**. זו קשת מקדקוד u לקדקוד v בגרף G ,

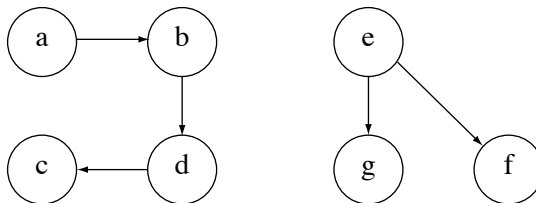
בתנאי שאף אחד משני קדקודים אלה אינו צאצא של הקדקוד האחר

בעץ הפורש / היער הפורש.

(1) לפניך גרף מכוון G , המכיל יותר מרק"ח (רכיב קשירות חזקה) אחד.



לאחר הפעלת אלגוריתם הסריקה לעומק (DFS) על הגרף המכוון G הנתון, התקבל היער הפורש שלפניך:



העתק למחברתך את הגרף G . הסתמך על היער הפורש הנתון וכתוב ליד

כל אחת מהקשתות של הגרף G אם היא:

— קשת עץ

— קשת אחורית

— קשת קדמית

— קשת חוצה

(2) לפניך טענה שחסר בה ביטוי במקום המסומן בקו.

אם לאחר הפעלת האלגוריתם סריקה לעומק (DFS) על גרף מכוון G נמצאת ב- G _____, אזי יש ב- G מעגל / מעגלים.

לפניך ארבעה ביטויים אפשריים להשלמת הטענה:

— קשת עץ

— קשת אחורית

— קשת קדמית

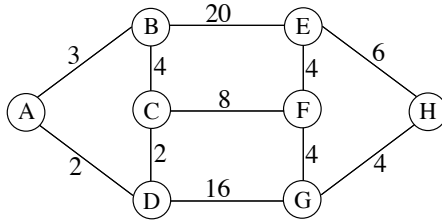
— קשת חוצה

העתק למחברתך את האפשרות המתאימה להשלמת הטענה, ונמק את קביעתך.

/המשך בעמוד 24/

10. בשאלה זו שני סעיפים א-ב שאינם תלויים זה בזה. ענה על שני הסעיפים.

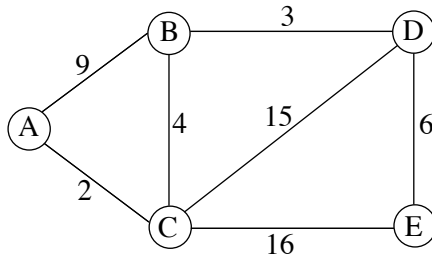
א. לפניך רשת $G = (V, E)$, V היא קבוצת הקדקודים ו- E היא קבוצת הקשתות:



השתמש באלגוריתם של Prim למציאת עץ פורש מינימלי וסרטט את העץ הפורש המינימלי המתקבל עבור הרשת הנתונה, החל מקדקוד C.

ב. (אין קשר לסעיף א.)

נתונה הרשת:



(1) השתמש באלגוריתם של Kruskal למציאת עץ פורש מינימלי וסרטט את

העץ הפורש המינימלי של הרשת הנתונה.

(2) תלמיד הציע למורה אלגוריתם למציאת עץ פורש מינימלי לרשת $G = (V, E)$,

כאשר V היא קבוצת הקדקודים ו- E היא קבוצת הקשתות של הרשת.

לפניך האלגוריתם של התלמיד.

צעד 1: צור גרף G_1 , שקבוצת הקדקודים שלו היא קבוצת הקדקודים של

הרשת G וקבוצת הקשתות שלו, E_T , היא קבוצה ריקה.

צעד 2: מייך את קשתות הרשת G בסדר עולה, על פי המשקלות המיוחסים

להן. תתקבל קבוצת קשתות ממוינות שנסמנה E^* .

צעד 3: לכל קשת (u, v) ב- E^* , החל מהקשת בעלת המשקל הקטן ביותר, בצע:

אם הוספת הקשת (u, v) אינה יוצרת מעגל ב- G_1 – הוסף את הקשת (u, v) ל- E_T .

אחרת – החזר את הגרף G_1 שהינו עץ פורש מינימלי.

צעד 4: החזר את G_1 שהינו עץ פורש מינימלי.

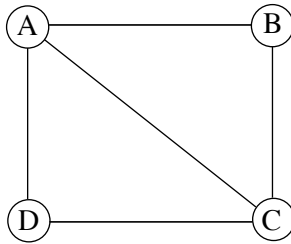
הפעל את האלגוריתם של התלמיד על הרשת הנתונה.

ענה על הסעיפים (i)-(iv).

- (i) סרטט במחברתך את G_1 שנוצר אחרי ביצוע צעד 1.
- (ii) רשום במחברתך את כל הקשתות ב- E^* אחרי ביצוע צעד 2.
- (iii) סרטט במחברתך כל אחד מהגרפים G_1 שהתקבלו במהלך צעד 3.
- (iv) סרטט במחברתך את העץ הפורש שהאלגוריתם החזיר.

(3) המורה השיבה לתלמיד שהאלגוריתם שהוא הציע שגוי, כי הפלט שלו

אינו בהכרח עץ פורש מינימלי. למשל, בעבור הרשת הזו:



אפשר לייחס משקלות לקשתות הרשת כך שאם נשתמש באלגוריתם של התלמיד לא יוחזר העץ הפורש המינימלי.

העתק למחברתך את הרשת ורשום משקלות לצדן של הקשתות, כך שהעץ שיוחזר על ידי האלגוריתם של התלמיד לא יהיה עץ פורש מינימלי.

11. בשאלה זו שני סעיפים א-ב שאינם תלויים זה בזה. ענה על שני הסעיפים.

א. נתונה בעיית תכנון לינארי:

$$\max \{z = kx_1 + 20x_2\}$$

בכפוף לאילוצים האלה:

$$-3x_1 + 4x_2 \leq 24$$

$$x_1 + 2x_2 \geq 0$$

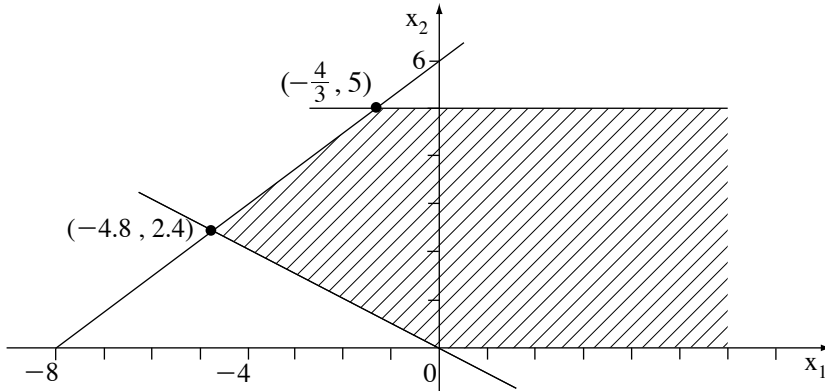
$$x_2 \leq 5$$

$$x_2 \geq 0$$

x_1 אינו מוגבל בסימן.

k הוא פרמטר.

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



לפניך ארבעה תת-סעיפים (1)-(4), שבכל אחד מהם נתון ערך מסוים של הפרמטר k .

(1) $k = 20$.

(2) $k = -20$.

(3) $k = -15$.

(4) $k = -15$, ומוסיפים את האילוץ: $x_1 \leq -5$.

התת-סעיפים אינם תלויים זה בזה.

בעבור כל אחד מהתת-סעיפים (1)-(4):

- אם הפתרון האופטימלי הוא יחיד, עליך למצוא את הפתרון האופטימלי היחיד, ואת הערך של פונקציית המטרה בפתרון זה.
- אם יש אין-סוף פתרונות אופטימליים, עליך לרשום את הפתרון האופטימלי הכללי לבעיה, ואת הערך של פונקציית המטרה בפתרון זה.
- אם הפתרון האופטימלי לא חסום, נמק את קביעתך.
- אם אין פתרון אפשרי, נמק את קביעתך.

ב. (אין קשר לסעיף א.)

נתונה בעיית תכנון לינארי:

$$\max \{z = ax_1 + 3x_2\}$$

בכפוף לאילוצים האלה:

$$x_1 + \frac{3}{8}x_2 \leq 3$$

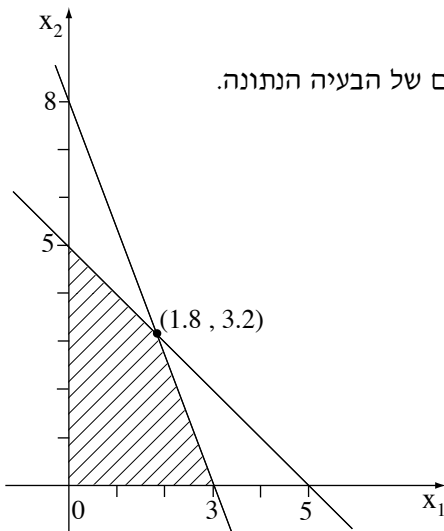
$$x_1 + x_2 \leq 5$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$a > 0$ הוא פרמטר של הבעיה.

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



לפניך ארבעה תחומים, iv-i, בעבור הפרמטר a :

$a < 3$ **i**

$a > 8$ **ii**

$3 < a < 8$ **iii**

$3 < a < 24$ **iv**

קבע בעבור איזה מבין התחומים iv-i, הפתרון $(1.8, 3.2)$ יהיה פתרון אופטימלי **יחיד**. העתק את התחום למחברתך, ונמק את קביעתך.

12. בשאלה זו תמישה סעיפים, א-ה, שאינם תלויים זה בזה. ענה על כל הסעיפים.
 א. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית התובלה:
 $x_{21} = 20$, $x_{11} = 80$

מקורות	יעדים			היצע
	1	2	3	
1	10	15	17	80
	80			
2	10	18	14	120
	20			
3	15	20	18	50
ביקוש	100	100	50	

העתק את הטבלה למחברתך, והשלם אותה לפי שיטת הפינה הצפונית מערבית.

- ב. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכים של $u_3, u_2, u_1, v_3, v_2, v_1$.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	10	15	12	80	0
	80				
2	10	18	10	100	3
		100			
3	20	10	14	50	-5
			50		
ביקוש	80	100	50		
v_j	10	15	19		

העתק את הטבלה למחברתך, והשלם אותה לפי שיטת הפינה הצפונית מערבית מתוך התחשבות בערכים של u_i ו- v_j ימים, כך שיתקבל פתרון בסיסי אפשרי.
 /המשך בעמוד 29/

ג. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכים של $u_1, u_2, u_3, v_1, v_2, v_3$.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	10 100	18	13 30	130	0
2	6 100	21	12	100	-4
3	11	16 50	14 100	150	1
ביקוש	200	50	130		
v_j	10	15	13		

האם הפתרון הוא אופטימלי? נמק את תשובתך.

ד. בטבלה שלפניך נתון פתרון אופטימלי לבעיית תובלה, ונתונים ערכים של $u_1, v_1, v_2, v_3, u_1, u_2, u_3$, שמתאימים לפתרון זה.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	10 20	25	16	20	10
2	10 30	22	14 20	50	10
3	18	20 40	20 20	60	16
ביקוש	50	40	40		
v_j	0	4	4		

(1) האם הפתרון הנתון הוא אופטימלי יחיד? נמק את תשובתך.

(2) בטבלה הנתונה בראש הסעיף הקסר 4 מהעלות ליחידה מכל מקור לכל יעד.

סרטט את הטבלה שתתקבל.

קבע איזה מארבעת ההיגדים $iv-i$ שלפניך הוא הנכון.

העתק אותו למחברתך, ונמק את בחירתך.

i הפתרון הנתון אינו פתרון אפשרי בעבור הבעיה שהתקבלה.

ii הפתרון הנתון הוא פתרון בסיסי אפשרי אך אינו אופטימלי בעבור

הבעיה שהתקבלה.

iii הפתרון הנתון הוא פתרון אופטימלי יחיד בעבור הבעיה שהתקבלה.

iv הפתרון הנתון הוא פתרון אופטימלי אך אינו פתרון אופטימלי יחיד בעבור

הבעיה שהתקבלה.

ה. בטבלה שלפניך נתון פתרון לא אופטימלי שהתקבל לאחר k איטרציות בעבור בעיית תובלה מסוימת, ונתונים ערכים של $u_3, u_2, u_1, v_4, v_3, v_2, v_1$.

מקורות	יעדים				היצע	u_i
	1	2	3	4		
1	5 150	2 80	4 -4	2 -6	230	0
2	2 70	3 4	5 80	3 -2	150	-3
3	7 7	4 7	3 200	3 340	540	-5
ביקוש	220	80	280	340		$Z = 3070$
v_j	5	2	8	8		

הפתרון המתואר בטבלה זו אינו אופטימלי כיוון שיש משתנים מחוץ לבסיס

שערך ה- $(c_{ij} - u_i - v_j)$ שלהם שלילי.

עליך לבצע איטרציה נוספת, כלומר איטרציה $k + 1$.

(1) מהו המשתנה שיוצא מהבסיס באיטרציה זו?

(2) סרטט במחברתך טבלה חדשה, ורשום בה את הפתרון שיתקבל לאחר

איטרציה זו.

מודלים חישוביים

אם למדת מסלול זה, ענה על שתיים מהשאלות 13-16 (לכל שאלה – 25 נקודות).

13. לפניך השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{a^n b^{3k+1} c^k \mid n > 0, k > 0\}$$

א. כתוב את המילה הקצרה ביותר בשפה L .

ב. בנה אוטומט מחסנית שיקבל את השפה L .

14. בשאלה זו שני סעיפים א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך השפה L מעל הא"ב $\{a, b\}$:

$$L = \{w \mid |w| > 0, w \text{ מופיע בכל המילה מספר זוגי של פעמים}\}$$

לדוגמא:

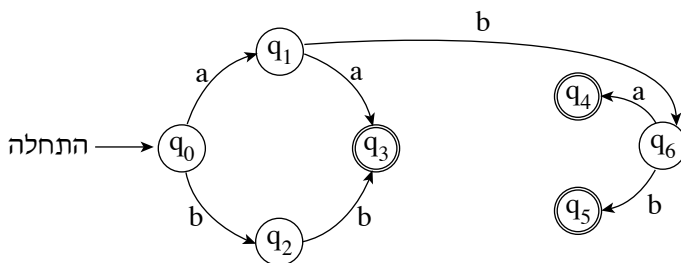
המילה $abaabbab$ שייכת לשפה.

המילה $abbabaa$ שייכת לשפה.

המילה $babbaba$ לא שייכת לשפה.

לפניך סרטוט חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L .

בסרטוט חסרים מעברים וסימני קלט.



הסרטוט מכיל את כל המצבים של האוטומט, ואת כל המצבים המקבלים.

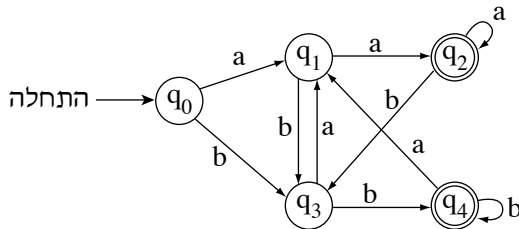
העתק למחברתך את הסרטוט, והשלם אותו כך שהאוטומט יקבל את השפה L .

עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים.

שים לב: אין להוסיף מצבים לאוטומט או להוריד ממנו מצבים.

ב. (אין קשר לסעיף א.)

לפניך אוטומט סופי דטרמיניסטי המקבל את שפה L מעל הא"ב $\{a, b\}$.



(1) קבע לכל אחת מארבע המילים (i)-(iv) שלפניך אם היא מתקבלת על ידי האוטומט. אם המילה מתקבלת על ידי האוטומט, רשום את המסלול המקבל בעבור מילה זו.

aaba (i)

bbaabb (ii)

abaa (iii)

bb (iv)

(2) מהי השפה L המוגדרת על ידי האוטומט?

15. לפניך השפה L מעל הא"ב $\{0, 1, 2\}$:

$$L = \{0^n 1^k 2 \mid n > k \geq 0\}$$

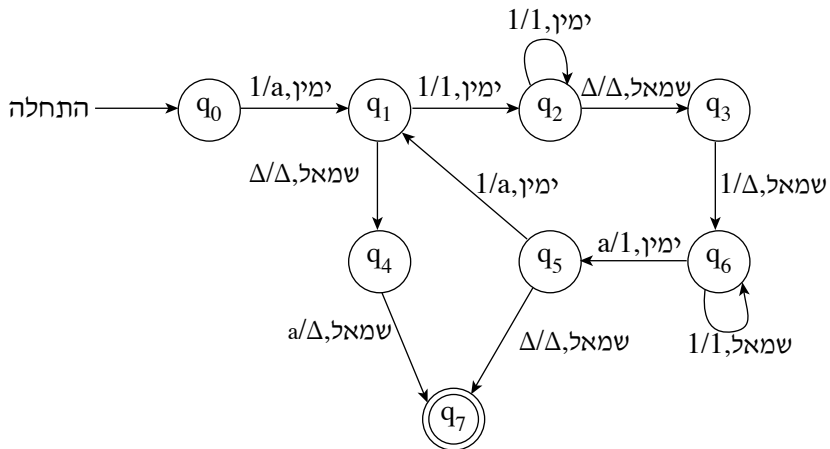
א. הוכח כי השפה L אינה רגולרית.

ב. נתונה השפה L_1 מעל הא"ב $\{1, 2\}$:

$$L_1 = \{1^n 2 \mid n \geq 0\}$$

מהי השפה $L_1 \cap L$? נמק.

16. לפניך מכונת טיורינג המחשבת פונקציה $f(x)$. המכונה מקבלת כקלט מספר x שלם וגדול מ-0, הרשום על הסרט כמספר אונרי על ידי x תווים של 1, ואחריהם הסימן Δ . המכונה רושמת את תוצאת החישוב של $f(x)$ על הסרט כמספר אונרי אחרי הסימן \vdash , ולאחר המספר מופיע הסימן Δ .



- א. מה יכיל הסרט לאחר חישוב $f(3)$? הראה את מסלול החישוב של המכונה. בכל שלב רשום את מצב הסרט, סמן היכן נמצא ראש המכונה, ורשום באיזה מהמצבים $q_0 - q_7$ המכונה נמצאת.
- ב. מה יכיל הסרט לאחר חישוב $f(5)$?
- ג. מה יכיל הסרט לאחר חישוב $f(6)$?
- ד. מהי הפונקציה $f(x)$ שהמכונה מחשבת?
- ה. הוסף למכונה מעבר/מעברים כדי שתחשב גם את $f(0)$. כתוב במחברתך את המעבר/מעברים שהוספת. לכל מעבר רשום: מאיזה מצב הוא יוצא, לאיזה מצב הוא מגיע ומה רשום עליו.

שים לב: תכנות מונחה עצמים מתחיל בעמוד 36.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב-Java, ענה על שתיים מהשאלות 17-20.
(לכל שאלה – 25 נקודות)

17. לפניך המחלקות **AA** ו-**BB** :

```
public class AA
{
```

```
    private String st;
```

```
    public AA()                { this.st = "excellent"; }

```

```
    public AA(String st)      { this.st = st; }

```

```
    public String getSt()     { return this.st; }

```

```
    public void setSt(String st) { this.st = st; }

```

```
    public String toString()  { return "st = " + this.st; }
}

```

```
public class BB extends AA
{
```

```
    private int num;
```

```
    public BB()                { super(); this.num = 1; }

```

```
    public BB(int num , String st) { super(st); this.num = Math.abs(num); }

```

```
    public int getNum()         { return this.num; }

```

```
    public void setNum(int num) { this.num = num; }

```

```
    public String toString()    { return super.toString() + " num = " + this.num;}
}

```

א. הגדר במחלקה **AA** פעולה בוליאנית בשם `isLike(Object obj)` המקבלת עצם `obj` מטיפוס `Object`. אם העצם `obj` הינו מטיפוס **AA** וגם תוכן המחרוזת `st` של `obj` זהה לתוכן המחרוזת `st` של העצם הנוכחי – הפעולה תחזיר `true`, אחרת – תחזיר `false`.

ב. הגדר במחלקה **BB** פעולה הדורסת את הפעולה שהגדרת בסעיף א.
אם העצם `obj` הינו מטיפוס **BB** וגם ערך התכונה `num` שלו זהה לערך התכונה `num` של העצם הנוכחי – הפעולה תחזיר `true`, אחרת – תחזיר `false`.

ג. לפניך קטע מפעולה ראשית:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.isLike(b)) System.out.println(a);
```

האם קטע התכנית תקין?

אם כן – מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה isLike תופעל – זו של AA או זו של BB.
אם לא – הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ד. לפניך קטע מפעולה ראשית:

```
AA aa = new AA();
BB bb = new BB(2, "excellent");
bb = aa;
if (bb.isLike(aa)) System.out.println(bb);
```

האם קטע התכנית תקין?

אם כן – מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה isLike תופעל – זו של AA או זו של BB.
אם לא – הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ה. כתוב פעולה חיצונית בשם longString המקבלת מערך של עצמים מטיפוס Object.

הפעולה מחזירה מחרוזת המורכבת משרשור התכונה st של עצמים מטיפוס AA במערך, באופן הזה:

– אם לעצם יש בק התכונה st, תשורשר המחרוזת שבתכונה st פעם אחת.

– אם לעצם יש גם התכונה num, המחרוזת שבתכונה st תשורשר num פעמים.

– אם אין במערך אף עצם מטיפוס AA, תוחזר מחרוזת ריקה.

18. לפניך פרויקט ובו המחלקות: **A**, **D**, **B** ו- **OopTest**:

```
public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        System.out.println("B(" + m1 + ", " + m2 + ") , #" + numB);
    }
}

public class D extends B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x)
    {
        super(x, x);
        this.d = d;
        numD++;
        System.out.println("D(" + d + ", " + x + ") , #" + numD);
    }

    public D(double d, int x, int y)
    {
        super(x , y);
        this.d = d;
        numD++;
        System.out.println("D(" + d + ", " + x + ", " + y + ") , #" + numD);
    }
}
}
```

```
public class A
{
    private static int numA = 0;
    private A a;
    private B b;

    public A(A a , B b)
    {
        this.a = a;
        this.b = b;
        numA++;
        System.out.println("A Constructor , #" + numA);
    }
}
```

```
public class OopTest
{
    public static void main(String [] args)
    {
        B w1 = new B(2, 3);
        B w2 = new D(1.5 , 6);
        B w3 = new D(2.3 , 8 , 9);
        A w4 = new A(null , w1);
        A w5 = new A(w4 , w3);
    }
}
```

כתוב מעקב אחר הפעולה main במחלקה **OopTest** , וכתוב את הפלט.

במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

19. חברה המפיצה לומדות פיתחה לומדה העוסקת בסדרות של מספרים שלמים. המערכת פותחה בשלבים.

בכל סדרה של מספרים מתייחסים אל:

- (1) האיבר הראשון בסדרה שמספרו הסידורי הוא 1.
- (2) האיבר שמספרו הסידורי בסדרה הוא n .
- (3) הדפסת n האיברים הראשונים בסדרה.

בשלב הראשון פותחו שתי מחלקות:

סדרה חשבונית (ASeq) – סדרה שבה ההפרש בין כל איבר לקודמו הוא ערך קבוע.
 סדרה הנדסית (GSeq) – סדרה שבה המנה בין כל איבר לקודמו היא ערך קבוע.

להלן קוד המחלקות שפותחו בשלב הראשון:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int theNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```



```

public void displayNElements(int n)
{
    System.out.print("The sequence elements: ");
    for (int i = 0; i < n-1; i++)
        System.out.print(this.theNElement(i+1) + " , ");
    System.out.println(this.theNElement(n));
}
}

```

```

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first , int product)
    {
        this.first = first;
        this.product = product;
    }

    public int theNElement(int n)
    {
        return this.first * (int)Math.pow(this.product , n-1);
    }

    public void displayNElements(int n)
    {
        System.out.print("The sequence elements: ");
        for (int i = 0 ; i < n-1; i++)
            System.out.print(this.theNElement(i+1) + " , ");
        System.out.println(this.theNElement(n));
    }
}

```

(שים לב: המשך השאלה בעמוד הבא.)

א. עקוב אחר קטע התכנית שלפניך. במעקב הצג את העצם שנבנה, את התכונות שלו ואת הפלט.

```
ASeq aSeq = new ASeq(2, 3);
System.out.println(aSeq.theNElement(4));
aSeq.displayNElements(5);
```

בשלב השני של הפיתוח הוחלט שמתאים לפתח מחלקה חדשה המתארת סדרה קבועה (**Sequence**), כך שהמחלקות **ASeq** ו-**GSeq** יירשו מן המחלקה החדשה. בסדרה קבועה מוגדר ערך האיבר הראשון, וכל יתר האיברים זהים לאיבר הראשון.

ב. השלם את הפיתוח של השלב השני באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים, ובהתאם להנחיות (i)-(ii):

(i) ממש באופן מלא את מחלקת העל **Sequence**. המחלקה צריכה להתייחס אל:

(1) האיבר הראשון בסדרה שמספרו הסידורי 1.

(2) האיבר שמספרו הסידורי בסדרה הוא n .

(3) הדפסת n האיברים הראשונים של הסדרה.

(ii) ממש מחדש את המחלקה **ASeq** כך שתירש מן המחלקה **Sequence**.

בשלב השלישי של הפיתוח הוחלט להרחיב את הפרויקט שכולל את שלוש המחלקות שפותחו בשלב השני (**Sequence**, **ASeq**, **GSeq**), כך שבעבור כל סדרה יהיה אפשר להפעיל פעולה המחשבת ומחזירה את סכום n האיברים הראשונים של הסדרה.

הנח כי המחלקה **GSeq** מומשה מחדש, כך שהיא יורשת מן המחלקה **Sequence**.

ג. בעבור כל אחת מן המחלקות **Sequence**, **ASeq**, **GSeq**, כתוב אם יש לעשות בה שינויים כך שהפרויקט יענה על דרישות הפיתוח של השלב השלישי באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים.

אם יש לעשות שינויים – פרט וממש אותם.

- בשלב הרביעי** של הפיתוח הוחלט לפתח פעולה סטטית check המקבלת ערך שלם n , ושני עצמים של סדרות: האחד מטיפוס **ASeq** והאחר מטיפוס **GSeq**. הפעולה מחשבת את סכום n האיברים הראשונים בכל אחת משתי הסדרות ומחזירה:
- את התו 'A' – אם סכום n האיברים הראשונים של הסדרה מטיפוס **ASeq** הוא הגדול מבין שני הסכומים.
 - את התו 'G' – אם סכום n האיברים הראשונים של הסדרה מטיפוס **GSeq** הוא הגדול מבין שני הסכומים.
 - את התו 'E' – אם סכום n האיברים הראשונים של שתי הסדרות שווה.
- ד.** ממש את הפעולה הסטטית check על פי הדרישות שהוגדרו בשלב הרביעי של הפיתוח.

20. מרפאה וטרינרית של חיות מחמד מרכזת מידע על הווטרינרים העובדים במרפאה ועל חיות המחמד המטופלות בה. מספר הווטרינרים העובדים במרפאה הוא לכל היותר 10, ומספר חיות המחמד המטופלות במרפאה הוא לכל היותר 500.

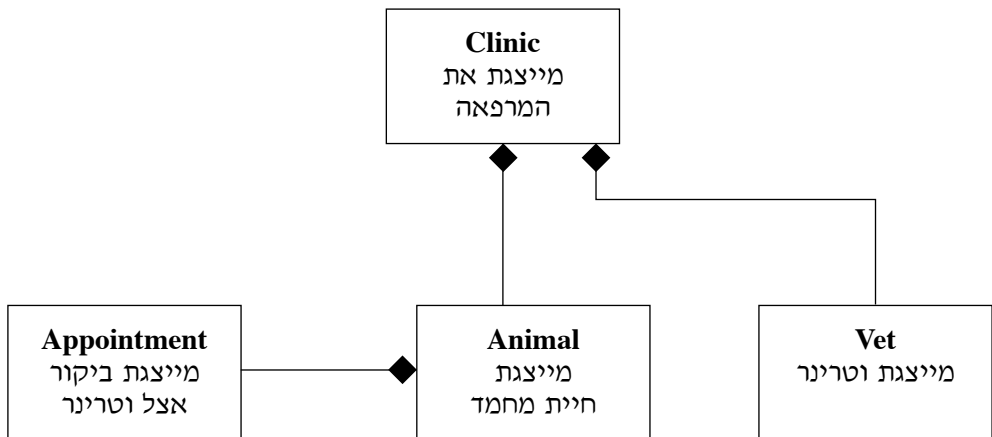
הקווים המנחים לניהול המידע הם:

- לכל וטרינר נשמר: מספר תעודת הזהות, שם, ותק בעבודה בשנים.
- לכל חייית מחמד נשמר: מספר הרישוי, שם, סוג (למשל: כלב, חתול או ארנב), גיל, פירוט של עד 50 הביקורים האחרונים במרפאה, מספר הביקורים השמורים.
- לכל ביקור במרפאה נשמר: מספר תעודת הזהות של הווטרינר שטיפל בחייית המחמד בביקור זה, ומחוזות של תווים המכילה קודים של טיפולים שחייית המחמד קיבלה באותו ביקור. קוד של טיפול הוא אות גדולה מתוך כל אותיות ה-ABC.

חלק מהפעולות שהמערכת יכולה לבצע הן:

- הפקת דוח של כל חיות המחמד מסוג מסוים (למשל כל הארנבים). לכל חיה הדוח יכול את מספר הרישוי שלה, את שמה ואת גילה.
- בתחילת כל שנה: עדכון הגיל של כל חייית מחמד, כלומר הגדלת הגיל ב-1, ועדכון הוותק של כל וטרינר, כלומר הגדלת הוותק ב-1.
- החזרת שם של וטרינר על פי מספר תעודת הזהות שלו.

לפניך תרשים של היררכיית המחלקות הנדרשות בעבור ניהול המידע של המרפאה:



הסימן —◆ בתרשים מייצג הכלה.

א.

בעבור כל מחלקה בתרשים, הגדר את התכונות ואת הפעולות שלה.
עליך להגדיר רק את הפעולות הנחוצות כדי לענות על הדרישות שתוארו בפתח לשאלה (שבעמוד 44) ובתרשים.

הנח שנתונות פעולות בונות, המקבלות פרמטר לכל תכונה, ופעולות מאחזרות (פעולות get), ואין צורך לכתוב אותן. לא נתונות פעולות קובעות (פעולות set).
בעבור כל תכונה – רשום את ההגדרה שלה ב- Java, ורשום את התיעוד שלה.
בעבור כל פעולה – רשום את הכותרת שלה ב- Java, ורשום תיעוד הכולל מה היא מקבלת ומה היא מחזירה. אין צורך לממש את הפעולה.

ב.

הנח כי המערכת פותחה בהתאם לתכנון שהצגת בסעיף א, וכל הפעולות הנתונות ואלו שהגדרת ממומשות.

במחלקה **Clinic** מוסיפים את הפעולה:

```
public void addAppointment (Animal p , String t , Vet v)
```

המקבלת חיית מחמד p, מחרוזת t של קודים של טיפולים שקיבלה החיה בביקור הנוכחי, ואת הווטרינר v שטיפל בה. הפעולה מוסיפה את הביקור לחיית המחמד. ממש באופן מלא את הפעולה.

הנח שמספר הביקורים הקודמים של החיה במרפאה קטן מ-50.

אם נוסף על הפעולות הנתונות ועל הפעולות שהגדרת בסעיף א אתה משתמש בפעולות אחרות, עליך לממש אותן באופן מלא ולציין בעבור כל פעולה באיזו מחלקה יש לממש אותה.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- `C#`, ענה על שתיים מהשאלות 21-24. (לכל שאלה – 25 נקודות)

21. לפניך המחלקות **AA** ו-**BB**:

```
public class AA
```

```
{
    private string st;

    public AA()                { this.st = "excellent"; }
    public AA(string st)      { this.st = st; }
    public string GetSt()     { return this.st; }
    public void SetSt (string st) { this.st = st; }
    public override string ToString() { return "st = " + this.st; }
}
```

```
public class BB : AA
```

```
{
    private int num;

    public BB() : base()      { this.num = 1; }
    public BB(int num, string st) : base(st) { this.num = Math.Abs(num); }
    public int GetNum()       { return this.num; }
    public void SetNum(int num) { this.num = num; }
    public override string ToString() { return base.ToString() + " num = "
                                        + this.num; }
}
```

א. הגדר במחלקה **AA** פעולה בוליאנית הניתנת לדריסה, בשם `IsLike (Object obj)`, המקבלת עצם `obj` מטיפוס `Object`. אם העצם `obj` הינו מטיפוס **AA** וגם תוכן המחרוזת `st` של `obj` זהה לתוכן המחרוזת `st` של העצם הנוכחי – הפעולה תחזיר `true`, אחרת – תחזיר `false`.

ב. הגדר במחלקה **BB** פעולה הדורסת את הפעולה שהגדרת בסעיף א. אם העצם `obj` הינו מטיפוס **BB** וגם ערך התכונה `num` שלו זהה לערך התכונה `num` של העצם הנוכחי – הפעולה תחזיר `true`, אחרת – תחזיר `false`.

ג. לפניך קטע מפעולה ראשית:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
```

האם קטע התכנית תקיין?

אם כן – מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה IsLike תופעל – זו של AA או זו של BB.
 אם לא – הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ד. לפניך קטע מפעולה ראשית:

```
AA aa = new AA();
BB bb = new BB(2, "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

האם קטע התכנית תקיין?

אם כן – מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה IsLike תופעל – זו של AA או זו של BB.
 אם לא – הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ה. כתוב פעולה חיצונית בשם LongString המקבלת מערך של עצמים מטיפוס Object.

הפעולה מחזירה מחרוזות המורכבת משרשור התכונה st של עצמים מטיפוס AA במערך, באופן הזה:

- אם לעצם יש רק התכונה st, תשורשר המחרוזות שבתכונה st פעם אחת.
- אם לעצם יש גם התכונה num, המחרוזות שבתכונה st תשורשר num פעמים.
- אם אין במערך אף עצם מטיפוס AA, תוחזר מחרוזת ריקה.

22. לפניך פרויקט ובו המחלקות **A, D, B** ו- **OopTest**:

```
public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        Console.WriteLine("B(" + m1 + ", " + m2 + ") , #" + numB);
    }
}

public class D : B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x) : base(x, x)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + ") , #" + numD);
    }

    public D(double d, int x, int y) : base(x, y)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + ", " + y + ") , #" + numD);
    }
}

/המשך בעמוד 49/
```



```
public class A
{
    private static int numA = 0;
    private A a;
    private B b;

    public A(A a , B b)
    {
        this.a = a;
        this.b = b;
        numA++;
        Console.WriteLine("A Constructor , #" + numA);
    }
}
```

```
public class OopTest
{
    public static void Main(string[] args)
    {
        B w1 = new B(2 , 3);
        B w2 = new D(1.5 , 6);
        B w3 = new D(2.3 , 8 , 9);
        A w4 = new A(null , w1);
        A w5 = new A(w4 , w3);
    }
}
```

כתוב מעקב אחר הפעולה Main במחלקה **OopTest** , וכתוב את הפלט.
במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

23. חברה המפיצה לומדות פיתחה לומדה העוסקת בסדרות של מספרים שלמים.

המערכת פותחה בשלבים.

עבור כל סדרה של מספרים מתייחסים אל:

(1) האיבר הראשון בסדרה שמספרו הסידורי הוא 1.

(2) האיבר שמספרו הסידורי בסדרה הוא n .

(3) הדפסת n האיברים הראשונים בסדרה.

בשלב הראשון פותחו שתי המחלקות:

סדרה חשבונית (ASeq) – סדרה שבה ההפרש בין כל איבר לקודמו הוא ערך קבוע.

סדרה הנדסית (GSeq) – סדרה שבה המנה בין כל איבר לקודמו היא ערך קבוע.

להלן קוד המחלקות שפותחו בשלב הראשון:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int TheNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```

```
public void DisplayNElements(int n)
{
    Console.Write("The sequence elements: ");
    for (int i = 0; i < n - 1; i++)
        Console.Write(this.TheNElement(i+1) + " , ");
    Console.WriteLine(this.TheNElement(n));
}
}

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first, int product)
    {
        this.first = first;
        this.product = product;
    }

    public int TheNElement(int n)
    {
        return this.first * (int)Math.Pow(this.product , n - 1);
    }

    public void DisplayNElements(int n)
    {
        Console.Write("The sequence elements: ");
        for (int i = 0; i < n - 1; i++)
            Console.Write(this.TheNElement(i+1) + " , ");
        Console.WriteLine (this.TheNElement(n));
    }
}
}
```

(שים לב: המשך השאלה בעמוד בא.)

א. עקוב אחר קטע התכנית שלפניך. במעקב הצג את העצם שנבנה, את התכונות שלו ואת הפלט.

```
ASeq aSeq = new ASeq(2 , 3);  
Console.WriteLine(aSeq.TheNElement(4));  
aSeq.DisplayNElements(5);
```

בשלב השני של הפיתוח הוחלט שמתאים לפתח מחלקה חדשה המתארת סדרה קבועה (Sequence), כך שהמחלקות **ASeq** ו-**GSeq** יירשו מן המחלקה החדשה. בסדרה קבועה מוגדר ערך האיבר הראשון, וכל יתר האיברים זהים לאיבר הראשון.

ב. השלם את הפיתוח של השלב השני באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים ובהתאם להנחיות (i)-(ii):

(i) ממש באופן מלא את מחלקת העל **Sequence**. המחלקה צריכה להתייחס אל:

- (1) האיבר הראשון בסדרה שמספרו הסידורי 1.
- (2) האיבר שמספרו הסידורי בסדרה הוא n.
- (3) הדפסת n האיברים הראשונים של הסדרה.

(ii) ממש מחדש את המחלקה **ASeq** כך שתירש מן המחלקה **Sequence**.

בשלב השלישי של הפיתוח הוחלט להרחיב את הפרויקט שכולל את שלוש המחלקות שפותחו בשלב השני (**Sequence**, **ASeq**, **GSeq**), כך שבעבור כל סדרה יהיה אפשר להפעיל פעולה המחשבת ומחזירה את סכום n האיברים הראשונים של הסדרה.

הנח כי המחלקה **GSeq** מומשה מחדש כך שהיא יורשת מן המחלקה **Sequence**.

ג. בעבור כל אחת מן המחלקות **Sequence**, **ASeq**, **GSeq**, כתוב אם יש לעשות בה שינויים כך שהפרויקט יענה על דרישות הפיתוח של השלב השלישי באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים. אם יש לעשות שינויים – פרט וממש אותם.

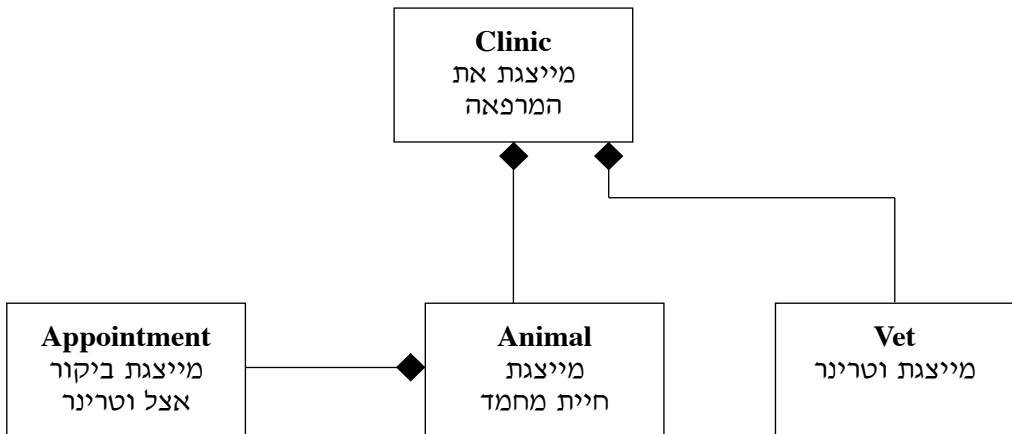
- בשלב הרביעי** של הפיתוח הוחלט לפתח פעולה סטטית Check המקבלת ערך שלם n , ושני עצמים של סדרות: האחד מטיפוס **ASeq** והאחר מטיפוס **GSeq**. הפעולה מחשבת את סכום n האיברים הראשונים בכל אחת משתי הסדרות ומחזירה:
- את התו 'A' – אם סכום n האיברים הראשונים של הסדרה מטיפוס **ASeq** הוא הגדול מבין שני הסכומים.
 - את התו 'G' – אם סכום n האיברים הראשונים של הסדרה מטיפוס **GSeq** הוא הגדול מבין שני הסכומים.
 - את התו 'E' – אם סכום n האיברים הראשונים של שתי הסדרות שווה.
- ד.** ממש את הפעולה הסטטית Check על פי הדרישות שהוגדרו בשלב הרביעי של הפיתוח.

24. מרפאה וטרינרית של חיות מחמד מרכזת מידע על הווטרינרים העובדים במרפאה ועל חיות המחמד המטופלות בה. מספר הווטרינרים העובדים במרפאה הוא לכל היותר 10, ומספר חיות המחמד המטופלות במרפאה הוא לכל היותר 500.

הקווים המנחים לניהול המידע הם:

- לכל וטרינר נשמר: מספר תעודת הזהות, שם, ותק בעבודה בשנים.
 - לכל חיות מחמד נשמר: מספר הרישוי, שם, סוג (למשל: כלב, חתול או ארנב), גיל, פירוט של עד 50 הביקורים האחרונים במרפאה, מספר הביקורים השמורים.
 - לכל ביקור במרפאה נשמר: מספר תעודת הזהות של הווטרינר שטיפל בחיית המחמד בביקור זה, ומחרוזת של תווים המכילה קודים של טיפולים שחיית המחמד קיבלה באותו ביקור. קוד של טיפול הוא אות גדולה מתוך כל אותיות ה-ABC.
- חלק מהפעולות שהמערכת יכולה לבצע הן:
- הפקת דוח של כל חיות המחמד מסוג מסוים (למשל כל הארנבים). לכל חיה הדוח יכלול את מספר הרישוי שלה, את שמה ואת גילה.
 - בתחילת כל שנה: עדכון הגיל של כל חיות מחמד, כלומר הגדלת הגיל ב-1, ועדכון הוותק של כל וטרינר, כלומר הגדלת הוותק ב-1.
 - החזרת שם של וטרינר על פי מספר תעודת הזהות שלו.

לפניך תרשים של היררכיית המחלקות הנדרשות בעבור ניהול המידע של המרפאה:



הסימן —◆ בתרשים מייצג הכלה.

א. בעבור כל מחלקה בתרשים, הגדר את התכונות ואת הפעולות שלה. עליך להגדיר רק את הפעולות הנחוצות כדי לענות על הדרישות שתוארו בפתיח לשאלה (שבעמוד 54) ובתרשים.

הנח שנתונות פעולות בונות, המקבלות פרמטר לכל תכונה, ופעולות מאחזרות (פעולות Get), ואין צורך לכתוב אותן. לא נתונות פעולות קובעות (פעולות Set). בעבור כל תכונה – רשום את ההגדרה שלה ב- C#, ורשום את התיעוד שלה. בעבור כל פעולה – רשום את הכותרת שלה ב- C#, ורשום תיעוד הכולל מה היא מקבלת ומה היא מחזירה. אין צורך לממש את הפעולה.

ב. הנח כי המערכת פותחה בהתאם לתכנון שהצגת בסעיף א, וכל הפעולות הנתונות ואלו שהגדרת ממומשות.

במחלקה **Clinic** מוסיפים את הפעולה:

```
public void AddAppointment (Animal p , string t , Vet v)
```

המקבלת חיית מחמד p, מחרוזת t של קודים של טיפולים שקיבלה החיה בביקור הנוכחי ואת הווטרין v שטיפל בה. הפעולה מוסיפה את הביקור לחיית המחמד. ממש באופן מלא את הפעולה.

הנח שמספר הביקורים הקודמים של החיה במרפאה קטן מ-50.

אם נוסף על הפעולות הנתונות ועל הפעולות שהגדרת בסעיף א אתה משתמש בפעולות אחרות, עליך לממש אותן באופן מלא ולציין בעבור כל פעולה באיזו מחלקה יש לממש אותה.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל
אין להעתיק או לפרסם אלא ברשות משרד החינוך