

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, — (25x2) — 50 נקודות
ומהן יש לענות על שתיים.
פרק שני – בפרק זה שאלות בארבעה מסלולים שונים — (25x2) — 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 כטייטה (ראשי פרקים, חישובים וכדומה).
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!
- ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.**
- בהצלחה!**
- /המשך מעבר לדף/

השאלות

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

פרק ראשון (50 נקודות)

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

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

1. כתוב פעולה חיצונית המקבלת שתי מחסניות $st1$, $st2$ המכילות מספרים שלמים וגדולים מ-0. הפעולה תחזיר את הסכום של זוג האיברים הסמוכים הקרוב ביותר לראש המחסנית $st1$, שסכומם גדול מהסכום של כל זוג איברים סמוכים במחסנית $st2$. אם אין זוג כזה – הפעולה תחזיר 0. הנח שבכל אחת מהמחסניות יש לפחות שני איברים. לדוגמה:

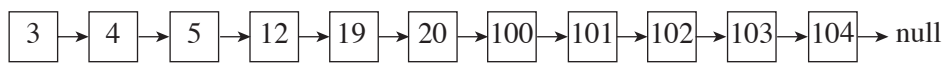
2	7
8	9
4	12
13	8
4	14
1	6
9	7
11	7
st2	st1

- סכום האיברים הסמוכים 9 ו-12 במחסנית $st1$ גדול יותר מסכום כל שני איברים סמוכים במחסנית $st2$, ולכן הפעולה תחזיר 21. שים לב: במחסנית $st1$ יש גם זוג איברים סמוכים שסכומם גדול יותר מ-21, אך הוא מופיע עמוק יותר במחסנית מזוג האיברים הסמוכים 9 ו-12. הערה: אין צורך לשמור על תוכן המקורי של המחסניות.

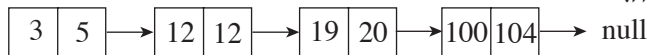
/המשך בעמוד 3/

2. L היא רשימה המכילה מספרים שלמים שונים זה מזה וממוינים בסדר עולה. **רשימת הטוחים** של L היא רשימה חדשה שנבנית באופן הזה: בעבור כל רצף של מספרים עוקבים ב-L יהיה ברשימת הטוחים איבר אחד שמכיל שני מספרים. מספר אחד הוא המספר הקטן ביותר ברצף, והמספר השני הוא המספר הגדול ביותר ברצף. רצף יכול להיות באורך 1 או יותר. אם הרצף הוא באורך 1, הוא מיוצג ברשימת הטוחים על ידי איבר ששני המספרים בו שווים.

לדוגמה, בעבור הרשימה L שלפניך:



רשימת הטוחים של L תהיה:



לפניך תיאור חלקי של המחלקה **RangeNode**, המייצגת איבר ברשימת הטוחים.

RangeNode	
private int from;	// המספר הקטן ביותר ברצף
private int to;	// המספר הגדול ביותר ברצף
public RangeNode(int from, int to)	

ממש ב-Java או ב-C# פעולה חיצונית שתקבל רשימה לא ריקה, המכילה מספרים שלמים שונים זה מזה וממוינים בסדר עולה, ותחזיר את רשימת הטוחים שלה.

כותרת הפעולה ב-Java היא:

```
public static List<RangeNode> createRangeList (List<Integer> sourceList)
```

כותרת הפעולה ב-C# היא:

```
public static List<RangeNode> CreateRangeList (List<int> sourceList)
```

הנח שלכל אחת מהתכונות במחלקה **RangeNode** יש פעולות get ו-set. אתה יכול להשתמש בפעולות אלה ובפעולה הבונה של המחלקה **RangeNode** בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

3. **שיום לב:** לשאלה זו שני נוסחים: אחד ב-Java (עמודים 4-5), ואחד ב-C# (עמודים 6-7). עבוד על פי השפה שלמדת.

לפותרים ב-Java

גולשים באתר "פטפטת" יכולים לשלוח הודעות זה לזה. לכל גולש מוקצית תיבת הודעות בגודל 100 קילו-בית. בכל תיבה מאוחסנות ההודעות שהגולש קיבל. לכל הודעה מוקצה מספר סידורי. ככל שההודעה חדשה יותר המספר הסידורי שלה גבוה יותר. כל הודעה מכילה את שם השולח, תוכן ההודעה, מספר סידורי, גודל ההודעה בקילו-בית. גודל ההודעה הוא מספר שלם. לתיבת ההודעות יש סל אשפה שהוא חלק מהתיבה. הודעות שנמצאות בתיבת ההודעות ואינן בסל האשפה נקראות הודעות פעילות. גולש יכול להעביר הודעות פעילות לסל האשפה או למחוק הודעות שנמצאות בסל האשפה. העברת הודעות לסל האשפה אינה גורמת לשינוי המספר הסידורי שלהן. הודעות שנמחקו מסל האשפה אינן קיימות יותר בתיבה. מחיקת הודעות אינה גורמת לשינוי המספר הסידורי של ההודעות שנשארו בתיבה.

לפניך חלק מממשק המחלקה **תיבת הודעות - MessageBox** :

MessageBox	
public MessageBox(String owner)	פעולה בונה המקבלת את שם בעל התיבה ויוצרת תיבה ריקה בגודל 100 קילו-בית.
public boolean addMessage(Message m)	פעולה המוסיפה הודעה חדשה m מטיפוס Message. אם אין די מקום בתיבה להודעה החדשה, יימחקו מסל האשפה ההודעות בעלות המספרים הסידוריים הנמוכים ביותר עד שיהיה די מקום להודעה החדשה. אם אין די מקום בתיבה ואין די הודעות למחוק בסל האשפה, ההודעה לא תתווסף לתיבה, ולא יימחקו הודעות מסל האשפה. הפעולה תחזיר true אם ההודעה הוספה לתיבה, אחרת הפעולה תחזיר false. הנחות: ל-m יש מספר סידורי מעודכן. גודל ההודעה קטן מגודל תיבת ההודעות.
public int removeFromBin()	פעולה המוחקת את ההודעה בעלת המספר הסידורי הקטן ביותר בסל האשפה ומחזירה את גודלה בקילו-בית.
public void emptyBin()	פעולה המוחקת את כל ההודעות מסל האשפה.
public int getActiveSize()	פעולה המחזירה את סך כל הגודל בקילו-בית שתופסות כל ההודעות הפעילות.
public int getBinSize()	פעולה המחזירה את סך כל הגודל בקילו-בית שתופסות כל ההודעות בסל האשפה.

- א. כתוב ב-Java את כותרת המחלקה **Message** המייצגת הודעה ב-**MessageBox**, ואת התכונות שלה. רשום תיעוד לכל תכונה.
- ב. כתוב ב-Java את כותרת המחלקה **MessageBox** ואת התכונות שלה. רשום תיעוד לכל תכונה.
- ג. ממש ב-Java את הפעולה `addMessage` המוצגת בממשק המחלקה **MessageBox**. אתה יכול להשתמש בפעולות האחרות של המחלקה **MessageBox** מבלי לממש אותן.
- אתה יכול להניח שלכל התכונות שרשמת בסעיף א השייכות למחלקה **Message** יש פעולות `get` ו-`set`.
- ד. לאחר שגולשים "פטפטנים" רבים ביקשו להגדיל את התיבה, הוחלט שפעם ביום תעבור המערכת על כל התיבות של הגולשים ותקטין או תגדיל את התיבות לפי הפעילות שלהם. תיבה שהתפוסה שלה יותר מ-80% תוגדל פי 2, ותיבה שהתפוסה שלה פחות מ-30% תוקטן בחצי.
- (1) האם לאור החלטה זו, תצטרך לשנות את תכונות המחלקה **MessageBox** שכתבת בסעיף ב? אם כן – הסבר מדוע, וכתוב את התכונות החדשות, ואם לא – הסבר מדוע.
- (2) האם לאור החלטה זו, תצטרך לשנות את הפעולה שכתבת בסעיף ג? אם כן – הסבר אילו שינויים יש לעשות, ואם לא – הסבר מדוע.

לפותרים ב- C#

גולשים באתר "פטפט" יכולים לשלוח הודעות זה לזה. לכל גולש מוקצית תיבת הודעות בגודל 100 קילו-בית. בכל תיבה מאוחסנות ההודעות שהגולש קיבל. לכל הודעה מוקצה מספר סידורי. ככל שההודעה חדשה יותר המספר הסידורי שלה גבוה יותר. כל הודעה מכילה את שם השולח, תוכן ההודעה, מספר סידורי, גודל ההודעה בקילו-בית. גודל ההודעה הוא מספר שלם. לתיבת ההודעות יש סל אשפה שהוא חלק מהתיבה. הודעות שנמצאות בתיבת ההודעות ואינן בסל האשפה נקראות הודעות פעילות. גולש יכול להעביר הודעות פעילות לסל האשפה או למחוק הודעות שנמצאות בסל האשפה. העברת הודעות לסל האשפה אינה גורמת לשינוי המספר הסידורי שלהן. הודעות שנמחקו מסל האשפה אינן קיימות יותר בתיבה. מחיקת הודעות אינה גורמת לשינוי המספר הסידורי של ההודעות שנשארו בתיבה. לפניך חלק מממשק המחלקה **תיבת הודעות – MessageBox** :

MessageBox	
public MessageBox(string owner)	פעולה בונה המקבלת את שם בעל התיבה ויוצרת תיבה ריקה בגודל 100 קילו-בית.
public bool AddMessage(Message m)	פעולה המוסיפה הודעה חדשה m מטיפוס Message. אם אין די מקום בתיבה להודעה החדשה, יימחקו מסל האשפה ההודעות בעלות המספרים הסידוריים הנמוכים ביותר עד שיהיה די מקום להודעה החדשה. אם אין די מקום בתיבה ואין די הודעות למחוק בסל האשפה, ההודעה <u>לא</u> תתווסף לתיבה, ו <u>לא</u> יימחקו הודעות מסל האשפה. הפעולה תחזיר true אם ההודעה הוספה לתיבה, אחרת הפעולה תחזיר false. <u>הנחות:</u> ל- m יש מספר סידורי מעודכן. גודל ההודעה קטן מגודל תיבת ההודעות.
public int RemoveFromBin()	פעולה המוחקת את ההודעה בעלת המספר הסידורי הקטן ביותר בסל האשפה ומחזירה את גודלה בקילו-בית.
public void EmptyBin()	פעולה המוחקת את כל ההודעות מסל האשפה.
public int GetActiveSize()	פעולה המחזירה את סך כל הגודל בקילו-בית שתופסות כל ההודעות הפעילות.
public int GetBinSize()	פעולה המחזירה את סך כל הגודל בקילו-בית שתופסות כל ההודעות בסל האשפה.

- א. כתוב ב- C# את כותרת המחלקה **Message** המייצגת הודעה ב- **MessageBox**, ואת התכונות שלה. רשום תיעוד לכל תכונה.
- ב. כתוב ב- C# את כותרת המחלקה **MessageBox** ואת התכונות שלה. רשום תיעוד לכל תכונה.
- ג. ממש ב- C# את הפעולה **AddMessage** המוצגת בממשק המחלקה **MessageBox**. אתה יכול להשתמש בפעולות האחרות של המחלקה **MessageBox** מבלי לממש אותן. אתה יכול להניח שלכל התכונות שרשמת בסעיף א השייכות למחלקה **Message** יש פעולות **Get** ו- **Set**.
- ד. לאחר שגולשים "פטפטנים" רבים ביקשו להגדיל את התיבה, הוחלט שפעם ביום תעבור המערכת על כל התיבות של הגולשים ותקטין או תגדיל את התיבות לפי הפעילות שלהם. תיבה שהתפוסה שלה יותר מ-80% תוגדל פי 2, ותיבה שהתפוסה שלה פחות מ-30% תוקטן בחצי.
- (1) האם לאור החלטה זו תצטרך לשנות את תכונות המחלקה **MessageBox** שכתבת בסעיף ב? אם כן – הסבר מדוע, וכתוב את התכונות החדשות, ואם לא – הסבר מדוע.
- (2) האם לאור החלטה זו תצטרך לשנות את הפעולה שכתבת בסעיף ג? אם כן – הסבר אילו שינויים יש לעשות, ואם לא – הסבר מדוע.

4. **שיום לב:** לשאלה זו שני נוסחים: אחד ב-Java (עמודים 8-9), ואחד ב-C# (עמודים 10-11). עבוד על פי השפה שלמדת.

לפותרים ב-Java

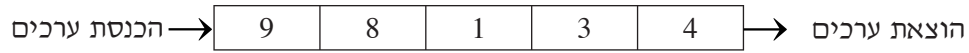
לפניך 2 פעולות חיצוניות הכתובות ב-Java .

```
public static void sod1(Queue<Integer> qu , Stack<Integer> st)
{
    if (!(qu.isEmpty()))
    {
        int x = qu.remove();
        sod1(qu , st);
        st.push(x);
        qu.insert(x);
    }
}
```

```
public static void sod2(Queue<Integer> qu , Stack<Integer> st)
{
    if (!(qu.isEmpty()))
    {
        int x = qu.remove();
        sod2(qu , st);
        int y = st.pop();
        qu.insert(x + y);
    }
}
```

/המשך בעמוד 9/

נתון התור myQueue מטיפוס `Queue<Integer>`:



ונתון קטע התכנית:

```
Stack<Integer> myStack = new Stack<Integer> ();
```

```
sod1(myQueue, myStack);
```

```
sod2(myQueue, myStack);
```

א. (1) עקוב אחר הביצוע של קטע התכנית ורשום את המעקב.

(2) ענה על התת-סעיפים (i)-(ii) על פי המעקב שרשמת.

(i) כתוב מה הכיל התור myQueue ומה הכילה המחסנית myStack

לאחר הזימון:
sod1(myQueue, myStack);

(ii) כתוב מה מכיל התור myQueue ומה מכילה המחסנית myStack

בסוף קטע התכנית.

ב. מה מבצעת הפעולה sod1 בעבור תור כלשהו לא ריק myQueue מטיפוס

`Queue<Integer>` ומחסנית myStack ריקה מטיפוס `Stack<Integer>`?

ג. מה מבצע קטע התכנית בעבור תור כלשהו לא ריק myQueue מטיפוס

`Queue<Integer>` ומחסנית myStack ריקה מטיפוס `Stack<Integer>`?

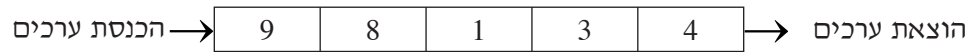
לפותרים ב- C#

לפניך 2 פעולות חיזוניות הכתובות ב- C# .

```
public static void Sod1 (Queue<int> qu , Stack<int> st)
{
    if (!(qu.IsEmpty()))
    {
        int x = qu.Remove();
        Sod1(qu , st);
        st.Push(x);
        qu.Insert(x);
    }
}

public static void Sod2 (Queue<int> qu , Stack<int> st)
{
    if (!(qu.IsEmpty()))
    {
        int x = qu.Remove();
        Sod2(qu , st);
        int y = st.Pop();
        qu.Insert(x + y);
    }
}
```

נתון התור myQueue מטיפוס Queue <int>



ונתון קטע התכנית:

```
Stack<int> myStack = new Stack<int> ();
```

```
Sod1(myQueue , myStack);
```

```
Sod2(myQueue , myStack);
```

א. (1) עקוב אחר הביצוע של קטע התכנית ורשום את המעקב.

(2) ענה על התת-סעיפים (i)-(ii) על פי המעקב שרשמת.

(i) כתוב מה הכיל התור myQueue ומה הכילה המחסנית myStack

לאחר הזימון:

```
Sod1 (myQueue , myStack);
```

(ii) כתוב מה מכיל התור myQueue ומה מכילה המחסנית myStack

בסוף קטע התכנית.

ב. מה מבצעת הפעולה Sod1 בעבור תור כלשהו לא ריק myQueue מטיפוס

Queue<int> ומחסנית myStack ריקה מטיפוס Stack<int>?

ג. מה מבצע קטע התכנית בעבור תור כלשהו לא ריק myQueue מטיפוס

Queue<int> ומחסנית myStack ריקה מטיפוס Stack<int>?

פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים שונים:

מערכות מחשב ואסמבלר, עמ' 12-17

מבוא לחקר ביצועים, עמ' 18-28

מודלים חישוביים, עמ' 29-32

תכנות מונחה עצמים ב-Java, עמ' 33-43; תכנות מונחה עצמים ב-C#, עמ' 45-55.

ענה רק על שאלות במסלול שלמדת.

מערכות מחשב ואסמבלר

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

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

א. במקטע הנתונים הוגדר מערך:

ARR DB 5 DUP(?)

לפניך 4 קטעים iv-i באסמבלר שמטרתם לאתחל ל-0 את כל תאי המערך ARR. עקוב בעזרת טבלת מעקב אחרי הביצוע של כל אחד מהקטעים iv-i שלפניך, וקבע אם הוא מבצע את הנדרש או אינו מבצע את הנדרש.

i		MOV	SI, 0
		MOV	CX, 4
	A1:	MOV	ARR[SI], 0
		INC	SI
		LOOP	A1
ii		MOV	CX, 5
		LEA	BX, ARR
		MOV	AL, 0
	A1:	MOV	[BX], AL
		INC	BX
	LOOP	A1	
iii		MOV	BX, 5
		DEC	BX
	A1:	MOV	ARR[BX], 0
		DEC	BX
		JNZ	A1
iv		MOV	DI, 0
	A1:	MOV	ARR[DI], 0
		INC	DI
		CMP	DI, 5
		JC	A1

/המשך בעמוד 13/

ב. (אין קשר לסעיף א.)

במחשנית מאוחסנים 10 מספרים בגודל מילה.
 מבין המספרים האלה יש להשאיר במחשנית רק את המספרים הזוגיים שקטנים מ-0.
 כמו כן יש לאחסן באוגר DX את כמות המספרים שנשארו במחשנית.
 הנח שיש לפחות מספר זוגי אחד שקטן מ-0.
 לפניך קטע תכנית המבצע זאת.
 מקטע התכנית הושמטו שורות במקומות המסומנים במספרים i-vi.
 רשום במחברתך את מספרי השורות המסומנות, וכתוב ליד כל אחד מהמספרים את שורת הפקודה החסרה, כדי שקטע התכנית יבצע את הנדרש.

		MOV	CX , 10
		MOV	BX , 0
	A1:	POP	AX
		MOV	DX , AX
i		_____	_____
		JC	A2
ii		_____	_____
		SHL	DX , 1
		JNC	A2
		MOV	[BX] , AX
iii		_____	_____
iv	A2:	_____	_____
		SUB	BX , 2
	A3:	MOV	AX , [BX]
v		_____	_____
		INC	CX
vi		_____	_____
		CMP	BX , 0
		JGE	A3
		MOV	DX , CX

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

א. הפעולה עצרת על מספר K שלם וגדול מ-0 מחשבת את מכפלת המספרים

מ-1 עד K , ומסומנת $K!$.

$$K! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (K - 1) \cdot K$$

לדוגמה: $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$

לפניך 4 קטעים iv-i באסמבלר, שמטרתם לחשב את ערך הביטוי $K!$ על פי ההגדרה

הכתובה למעלה, ולהציב את התוצאה ב- AX .

במקטע הנתונים הוגדר:

4 DW K

עקוב בעזרת טבלת מעקב אחר ביצוע כל אחד מהקטעים iv-i שלפניך, וקבע אם הוא

מבצע את הנדרש או אינו מבצע את הנדרש.

i		XOR	CX , CX
		MOV	AX , 1
		MOV	CX , K
	A1:	MUL	CX
		LOOP	A1
ii		MOV	AX , 1
		MOV	CX , 1
	A1:	MUL	CX
		INC	AX
		CMP	AX , K
		JBE	A1
iii		XOR	CX , CX
		MOV	AX , K
		INC	CX
	A1:	MUL	CX
		DEC	AX
		LOOP	A1
iv		MOV	AX , 1
		MOV	CX , K
	A1:	MUL	CX
		DEC	CX
		CMP	CX , 0
		JNZ	A1

/המשך בעמוד 15/

ב. (אין קשר לסעיף א.)

לפניך 10 הוראות באסמבלר, שאינן קשורות זו לזו.

בחלק מההוראות נפלה שגיאה.

העתק למחברתך רק את ההוראות השגויות, והסבר מהי השגיאה בכל אחת מהן.

- MOV BX , AL
- MOV [SI] , SI
- MOV AL , 100H
- AND BX , 256
- DEC DI
- SUB 5 , BX
- MOV [SI] , [BX]
- MOV [SI + DI] , AL
- ROL BX , CL
- DIV AX

7. נגדיר מערך **ממוין למחצה** באופן הזה:

מערך באורך זוגי המכיל מספרים גדולים מ-0 ושונים זה מזה, יש בו רצף של איברים סמוכים ממוין בסדר עולה, ואורך הרצף הזה לפחות מחצית מאורך המערך.
לדוגמה:

המערך בגודל 10 שלפניך הוא **ממוין למחצה**.

הרצף של איברים הממוינים בסדר עולה נמצא בין מציין 2 למציין 6 (מסומן באפור) ואורכו 5.

0	1	2	3	4	5	6	7	8	9
8	4	3	5	10	19	35	7	1	9

במקטע הנתונים הוגדרו הנתונים כך:

LOW DB ?
HIGH DB ?
ARR DB 100 DUP(?)

א. כתוב באסמבלר שגרה (פרוצדורה) בשם CHECK, שתקבל את המערך ARR ושני מספרים שלמים LOW ו-HIGH (קטן מ-HIGH). שני המספרים הם מציינים (אינדקסים) של איברים במערך.

השגרה (פרוצדורה) תבדוק אם איברי המערך, מהאיבר שהמציין שלו הוא LOW ועד האיבר שהמציין שלו הוא HIGH (כולל LOW ו-HIGH), ממוינים בסדר עולה.

אם כן – השגרה (פרוצדורה) תאחסן 1 באוגר AH, אחרת – היא תאחסן 0 ב-AH.

הנח שאיברי המערך ARR גדולים מ-0 ושונים זה מזה.

ב. כתוב באסמבלר קטע תכנית שיבדוק אם המערך ARR הוא מערך **ממוין למחצה**.

אם כן – קטע התכנית יאחסן 1 באוגר AH, אחרת – הוא יאחסן 0 ב-AH. עליך להשתמש בשגרה (פרוצדורה) CHECK שכתבת בסעיף א.

8. במקטע הנתונים הוגדרו הנתונים כך:

```
T   DB  1, 2, 3, 4, 5, 6
T1  DB  6 DUP (0)
T2  DB  6 DUP (0)
```

לפניך קטע תכנית באסמבלר:

```
      LEA    BX, T
      LEA    DI, T1
      LEA    SI, T2
      MOV    CX, 6
SH:   MOV    AL, [BX]
      MOV    AH, AL
      AND    AH, 1
      JZ     E
      MOV    [DI], AL
      INC    DI
      JMP    NEXT
E:    MOV    [SI], AL
      INC    SI
NEXT: INC    BX
      LOOP  SH
```

- א. עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית, ורשום את התוכן של המערכים T1 ו-T2 לאחר ביצועו.
- ב. מה מבצע קטע התכנית?
- ג. אם תוחלף ההוראה LOOP בהוראה LOOPE, האם תשתנה תשובתך לסעיף א? אם כן – הסבר מה ישתנה, אם לא – הסבר מדוע.

מבוא לחקר ביצועים

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

9. נתונה בעיית תכנון לינארי:

$$\text{Min } \{z = x_1 - x_2\}$$

בכפוף לאילוצים האלה:

$$x_1 + x_2 \geq 2$$

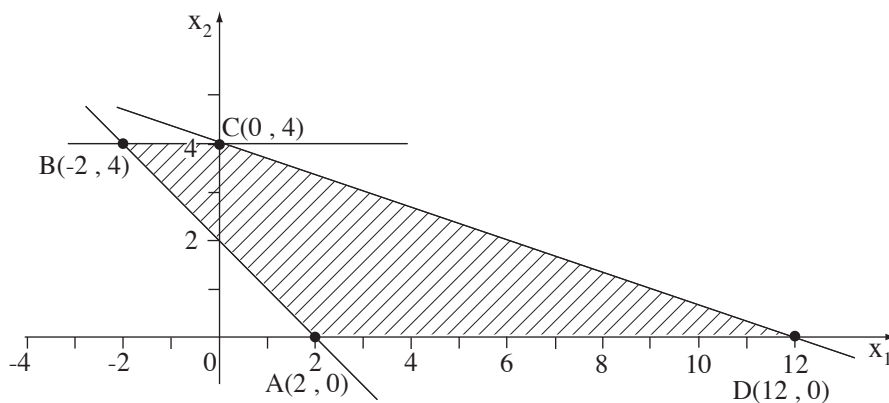
$$x_1 + 3x_2 \leq 12$$

$$x_2 \leq 4$$

$$x_2 \geq 0$$

x_1 אינו מוגבל בסימן.

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



כל אחד מהסעיפים א-ד שלפניך מתייחס לבעיית התכנון הלינארי הנתונה. הסעיפים אינם תלויים זה בזה. ענה על כל הסעיפים.

א. מהו הפתרון האופטימלי של הבעיה הנתונה? נמק את תשובתך.

ב. משנים רק את פונקציית המטרה של הבעיה שבתחילת השאלה ל- $z = x_1 + x_2$.

(1) הראה שלבעיה יש אין-סוף פתרונות אופטימליים.

(2) רשום את הפתרון האופטימלי הכללי לבעיה, ואת ערכה של פונקציית המטרה

בתחום הפתרונות האופטימליים.

ג. משנים את פונקציית המטרה של הבעיה שבתחילת השאלה ל- $z = x_1 + 2x_2$ ומבטלים את האילוץ $x_2 \geq 0$.

(1) סרטט את תחום הפתרונות האפשריים שהתקבל לאחר השינוי.

(2) בצע את הצעדים האלה:

צעד 1: הוסף לסרטוט שבמחברתך את היטל הגובה של פונקציית המטרה בעבור $z = 8$.

חשב את שיעורי נקודות החיתוך של היטל זה עם הצירים x_1 ו- x_2 , וסמן אותן על הסרטוט.

צעד 2: הוסף לסרטוט שבמחברתך את היטל הגובה של פונקציית המטרה בעבור $z = 4$.

חשב את שיעורי נקודות החיתוך של היטל זה עם הצירים x_1 ו- x_2 , וסמן אותן על הסרטוט.

צעד 3: סמן בסרטוט שבמחברתך, באמצעות חץ, את כיוון הירידה של פונקציית המטרה.

(3) לפניך ההיגדים i-iv, שרק אחד מהם נכון.

על סמך הצעדים 1-3 שביצעת בתת-סעיף ג(2), קבע איזה מבין ההיגדים הוא הנכון, העתק אותו למחברתך, ונמק את קביעתך.

i הפתרון האופטימלי הוא יחיד.

ii יש אין-סוף פתרונות אופטימליים.

iii הפתרון האופטימלי לא חסום.

iv אין פתרון אופטימלי.

ד. משנים רק את פונקציית המטרה של הבעיה שבתחילת השאלה ל- $z = a(x_1 - 2x_2)$.

נתון ש- $a > 0$ והערך המקסימלי שמקבלת פונקציית המטרה בתחום הפתרונות האפשריים הוא 120.

חשב את הערך המינימלי שמקבלת פונקציית המטרה בתחום הפתרונות האפשריים.

10. בשאלה זו ששעה סעיפים, א-ו, שאינם תלויים זה בזה. ענה על כל הסעיפים.
 א. לפניך בעיית תובלה ובה 3 מקורות ו-3 יעדים. העלויות ליחידה מכל מקור לכל יעד נתונות בטבלה שלפניך.

מקורות	יעדים			היצע
	1	2	3	
1	18	19	16	180
2	14	15	18	a
3	15	20	17	100
ביקוש	120	150	150	

בבעיה הנתונה אפשר להפעיל את שיטת הפינה הצפונית-מערבית כדי למצוא פתרון אפשרי לבעיית התובלה הנתונה, ללא הוספת יעד דמה או מקור דמה. מה צריך להיות ערכו של המשתנה a שבטבלה? נמק את תשובתך.

- ב. בטבלה שלפניך נתון חלק מפתרון בסיסי לבעיית תובלה: $x_{11} = 90$, $x_{12} = 10$.

מקורות	יעדים			היצע
	1	2	3	
1	10 90	15 10	17	100
2	10	18	14	110
3	15	20	18	100
ביקוש	90	120	100	

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

ג. בטבלה שלפניך נתון חלק מפתרון אפשרי לבעיית תובלה, ונתונים הערכים של

$$u_1, u_2, u_3, v_1, v_2, v_3$$

מקורות	יעדים			היצע	u_i
	1	2	3		
1	12 200	15	17	200	2
2	10	18 100	14	100	0
3	20	10	18 100	150	-8
ביקוש	200	150	100		
v_j	10	18	26		

העתק את הטבלה למחברתך, והשלם אותה לפי שיטת הפינה הצפונית-מערבית תוך התחשבות בערכים של u_i ו- v_j ים, כך שיתקבל פתרון בסיסי אפשרי.

ד. בטבלה שלפניך נתון פתרון בסיסי לבעיית תובלה, ונתונים הערכים של $u_1, u_2, u_3, v_1, v_2, v_3$. האם הפתרון הוא אופטימלי? נמק את תשובתך.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	14	15 150	17 30	180	0
2	10 100	8 0	14	100	-7
3	15	20	18 100	100	1
ביקוש	100	150	130		
v_j	17	15	17		

ה. בטבלה שלפניך נתון פתרון לבעיית תובלה, ונתונים הערכים של

$u_3, u_2, u_1, v_3, v_2, v_1$, שמתאימים לפתרון זה.

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

לפניך ההיגדים $iv-i$ שרק אחד מהם נכון. העתק למחברתך את ההיגד הנכון, ונמק את בחירתך.

- i** הפתרון הנתון איננו פתרון אפשרי.
- ii** הפתרון הנתון הוא פתרון בסיסי אפשרי אך איננו אופטימלי.
- iii** הפתרון הנתון הוא פתרון אופטימלי יחיד.
- iv** הפתרון הנתון הוא פתרון אופטימלי אך איננו פתרון אופטימלי יחיד.

1. נתונה בעיית תובלה.

בטבלה שלפניך מוצגים נתוני הבעיה, כולל מחירי התובלה של יחידת מוצר מהמקורות ליעדים.

מקורות	יעדים			היצע
	1	2	3	
1	20		34	100
2	10		14	120
3	15	20	18	130
4	25	25	32	40
ביקוש	130	130	130	

ממקורות 1 ו-2 אסור לספק ליעד 2.

מה צריכות להיות העלויות c_{12} , c_{22} (M מוגדר כמספר גדול מאוד)?

לפניך תשובות iv-i שרק אחת מהן נכונה. העתק למחברתך את התשובה הנכונה.

i $c_{12} = M$, $c_{22} = 0$

ii $c_{12} = 0$, $c_{22} = 0$

iii $c_{12} = 0$, $c_{22} = M$

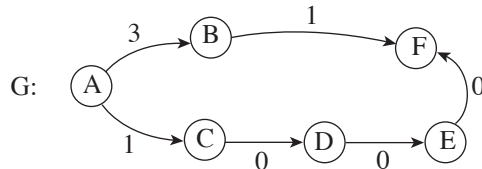
iv $c_{12} = M$, $c_{22} = M$

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

א. לפניך גרף מכוון $G = (V, E)$.

V היא קבוצת הקדקודים, ו- E היא קבוצת הקשתות.

פונקציית המשקל $W: E \rightarrow \mathbb{R}^+$ קובעת משקל (מספר) לכל קשת בגרף G .



(1) רשום במחברתך את רשימת הקדקודים במסלול הקצר ביותר מקדקוד A לקדקוד F (כולל A ו-F).

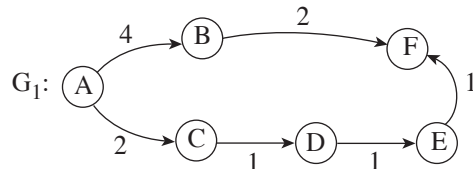
(2) יהי גרף מכוון $G_1 = (V, E)$, המכיל את אותם קדקודים כמו הגרף G ואת

אותן קשתות, ופונקציית המשקל $W_1: E \rightarrow \mathbb{R}^+$ מוגדרת באופן הזה:

לכל קשת e ב- G_1 , $W_1(e) = W(e) + t$, כאשר W היא פונקציית המשקל

של G ו- t מספר שלם וגדול מ-0.

בעבור $t = 1$ מתקבל הגרף G_1 שלפניך:



רשום במחברתך את רשימת הקדקודים במסלול הקצר ביותר מקדקוד A לקדקוד

F (כולל A ו-F) בגרף G_1 .

(3) לפניך משפט שיש להשלימו:

רשימת הקדקודים במסלול הקצר ביותר מקדקוד A לקדקוד F בגרף G , ורשימת

הקדקודים במסלול הקצר ביותר מקדקוד A לקדקוד F בגרף G_1 הן: _____.

לפניך שלוש אפשרויות i-iii להשלמת המשפט.

העתק למחברתך את ההשלמה הנכונה, ונמק את בחירתך.

i שוות לכל t .

ii שונות לכל t .

iii יכולות להיות שוות או שונות, תלוי בערכו של t .

ב. (אין קשר לסעיף א.)

- הגרף G הוא גרף קשיר ולא מכוון המוגדר על ידי $G = (V, E)$.
 V היא קבוצת הקדקודים, ו- E היא קבוצת הקשתות.
 פונקציית המשקל $W: E \rightarrow \mathbb{R}^+$ קובעת משקל (מספר) לכל קשת בגרף G .
 יהיו s ו- t קדקודים בגרף G , ו- U תת-קבוצה של V .
 לפניך תיאור אלגוריתם המוצא מסלול מ- s ל- t בגרף G , המבקר במספר מינימלי של קדקודים ב- U .
 באלגוריתם חסרים ארבעה ביטויים המסומנים (1) - (4).

תיאור האלגוריתם:

צעד 1: נבנה את הגרף $G_1 = (V_1, E_1)$ כאשר:

$$V_1 \leftarrow V$$

$$E_1 \leftarrow E$$

ופונקציית המשקל $W: E_1 \rightarrow \{0, 1, 2\}$, המוגדרת באופן הזה:

(1) $W(x, y) =$ כאשר הקדקודים x ו- y לא שייכים לקבוצה U .

(2) $W(x, y) =$ כאשר רק אחד הקדקודים x או y שייך לקבוצה U .

(3) $W(x, y) =$ כאשר הקדקודים x ו- y שייכים לקבוצה U .

צעד 2: נפעיל את אלגוריתם (4) על הגרף G_1 החל מקדקוד s .

צעד 3: התקבל המסלול המבוקש ואפשר לשחזר אותו מתוך מערך

ה- $parent$ שנבנה באלגוריתם החל מ- s .

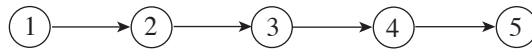
העתק למחברתך את הטבלה שלפניך, ורשום בה את הביטויים החסרים.

	ביטוי (1)
	ביטוי (2)
	ביטוי (3)
	ביטוי (4)

12. לפניך ארבע הגדרות i-iv:

i גרף מכוון בעל n קדקודים נקרא **שרוך מכוון** אם אין בו מעגל ואם דרגת הכניסה ודרגת היציאה של כל אחד מ- $n-2$ קדקודים היא 1, יש בו קדקוד אחד שדרגת הכניסה שלו היא 0 ודרגת היציאה שלו היא 1, וקדקוד אחד שדרגת הכניסה שלו היא 1 ודרגת היציאה שלו היא 0.

לפניך **שרוך מכוון**:



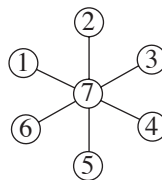
ii גרף לא מכוון נקרא **שרוך** אם הוא קשיר ודרגת כל קדקוד היא 2, פרט לשני קדקודים שדרגתם היא 1.

לפניך **שרוך**:



iii גרף לא מכוון בעל n קדקודים נקרא **גרף כוכב** אם הוא קשיר, הדרגה של אחד הקדקודים היא $n-1$, והדרגה של כל אחד מהקדקודים האחרים היא 1. הקדקוד שדרגתו $n-1$ נקרא **קדקוד מרכזי**.

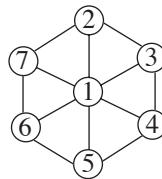
לפניך **גרף כוכב** בעל 7 קדקודים:



iv גרף לא מכוון בעל n קדקודים המסומנים 1, 2, ..., n , $n \geq 4$, נקרא **גרף גלגל** אם הדרגה של הקדקוד המסומן 1 היא $n-1$, כלומר הוא מקושר לכל אחד מן הקדקודים האחרים. כל קדקוד i , $3 \leq i \leq n-1$ מקושר לקדקוד 1, לקדקוד $i+1$ ולקדקוד $i-1$. כמו כן יש קשת בין הקדקודים 2 ו- n .

הקדקוד שדרגתו $n-1$ נקרא **קדקוד מרכזי**.

לפניך **גרף גלגל** בעל 7 קדקודים:



לפניך ארבעה סעיפים א-d שמתייחסים להגדרות $i-v$ שבעמוד הקודם. הסעיפים אינם תלויים זה בזה. ענה על כל הסעיפים.

א. הגרף שלפניך הוא שרוך מכוון.



(1) סרטט את ייצוגו של הגרף באמצעות רשימת סמיכויות.

(2) רשום את מטריצת הסמיכויות המייצגת את הגרף.

ב. לפניך אלגוריתם הבודק אם גרף קשיר הוא גרף כוכב. האלגוריתם פועל על גרף לא

מכוון $G = (V, E)$ בעל n קדקודים המיוצג באמצעות מטריצת סמיכויות.

באלגוריתם חסרים שני ביטויים המסומנים (1) ו-(2).

תיאור האלגוריתם:

צעד 1: בדוק אם לכל קדקוד v בגרף, מספר ה-1ים בשורה v במטריצת

הסמיכויות הוא (1), פרט לקדקוד אחד u שמספר ה-1ים בשורה u

במטריצת הסמיכויות הוא (2).

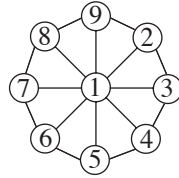
צעד 2: אם כן – החזר "גרף כוכב", אחרת – החזר "לא גרף כוכב".

העתק למחברתך את הטבלה שלפניך, ורשום בה את הביטויים החסרים.

	ביטוי (1)
	ביטוי (2)

(שים לב: המשך השאלה בעמוד הבא.)

ג. לפניך גרף גלגל בעל 9 קודקודים:



- (1) הפעל את האלגוריתם DFS על הגרף החל מהקדקוד המרכזי. האם העץ הפורש DFS שיתקבל הוא שרוך?
- (2) אם נתחיל את האלגוריתם DFS מקדקוד אחר של הגרף, שאינו הקדקוד המרכזי, האם העץ הפורש DFS שיתקבל הוא שרוך? נמק את תשובתך.

- ד. נתון גרף לא מכוון $G = (V, E)$ מלא (שלם) בן n קדקודים. $n > 3$. רוצים לדעת אם קיימת הרצת DFS כך שהעץ הפורש DFS המתקבל ממנה הוא עץ שבו דרגת השורש (הקדקוד הראשון שנסרק) היא 2 בדיוק. אם קיימת הרצה כזו – כתוב אותה. אחרת – הסבר מדוע אין הרצה כזו.

מודלים חישוביים

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

13. לפניך 3 שפות $L_3 - L_1$ מעל הא"ב $\{a, b\}$:

$$L_1 = \{a^n b^{n-1} \mid n \geq 1\}$$

$$L_2 = \left\{ a^n b^k \mid \begin{array}{l} n \geq 0, k \geq 0, \text{ שארית החלוקה של } n \text{ ב- } 2 \text{ שונה} \\ \text{משארית החלוקה של } k \text{ ב- } 2 \end{array} \right\}$$

$$L_3 = \{a^k b^{2m} \mid k \geq 0, m \geq 0\}$$

א. לכל אחת משלוש השפות $L_3 - L_1$, קבע אם היא רגולרית או אינה רגולרית.

אם השפה רגולרית – בנה אוטומט סופי דטרמיניסטי שיקבל אותה.

אם אינה רגולרית – נמק את קביעתך.

ב. לפניך השפות $L_5 - L_4$ מעל הא"ב $\{a, b\}$:

$$L_4 = \{a^n b^{n-1} \mid n \text{ הוא אי-זוגי}, n \geq 1\}$$

$$L_5 = L_1 \cdot R(L_4)$$

מהי השפה L_5 ?

14. לפניך השפה L מעל הא"ב $\{0, \$\}$:

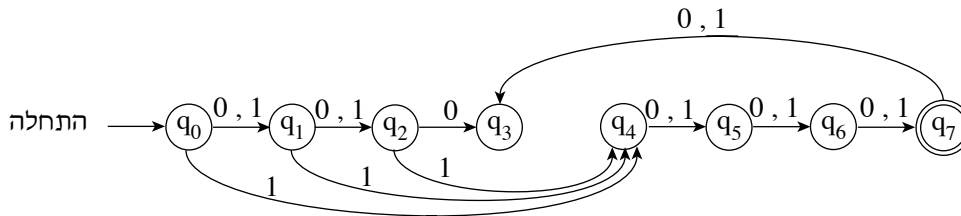
$$L = \left\{ 0^3 \$ 0^{i_1} \$ 0^{i_2} \$ \dots 0^{i_k} \$ \mid \begin{array}{l} k \geq 1 \\ \text{לכל } m \text{ בין } 1 \text{ ל- } k : i_m \geq 0 \\ \text{ו- } i_m \text{ מתחלק ב- } 3 \text{ ללא שארית} \end{array} \right\}$$

א. כתוב את המילה הקצרה ביותר בשפה L .

ב. בנה אוטומט סופי דטרמיניסטי שיקבל את השפה L .

15. בשאלה זו שני סעיפים, א-ב שאין קשר ביניהם. ענה על שני הסעיפים.

א. לפניך אוטומט סופי לא דטרמיניסטי:



(1) קבע לכל אחת משלוש המילים i - iii שלפניך אם היא מתקבלת על ידי

האוטומט.

אם המילה מתקבלת על ידי האוטומט, רשום מסלול מקבל בעבור מילה זו.

001001 i

01010 ii

0101 iii

(2) מהו האורך המינימלי של מילה המתקבלת על ידי האוטומט?

תן דוגמה למילה באורך זה.

(3) מהו האורך המקסימלי של מילה המתקבלת על ידי האוטומט?

תן דוגמה למילה באורך זה.

(4) מהי השפה המוגדרת על ידי האוטומט?

ב. (אין קשר לסעיף א.)

לפניך השפה L מעל הא"ב $\{a, b\}$:

$$L = \left\{ w \mid \begin{array}{l} w \text{ היא מילה שבה מופיע הרצף } aaa \text{ בדיוק פעם אחת,} \\ \text{אין בה מופעים של הרצף } aa \text{ פרט לאלה שב-} aaa, \\ \text{ואין בה רצף של } a\text{-ים שאורכו גדול מ-} 3. \end{array} \right.$$

לדוגמה:

המילה $ababbaaabbbba$ שייכת ל- L .

המילה $aaabbbabbab$ שייכת ל- L .

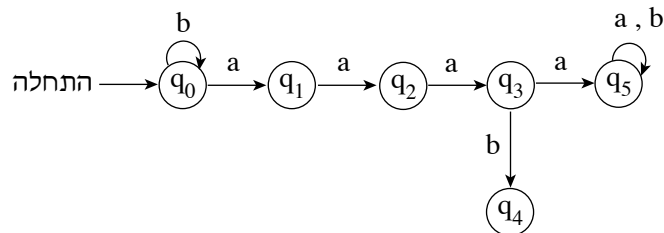
המילה $bababbbab$ לא שייכת ל- L .

המילה $bbaaabaab$ לא שייכת ל- L .

המילה $aaababaaabbb$ לא שייכת ל- L .

המילה $aaaa$ לא שייכת ל- L .

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



הסרטוט מכיל את כל המצבים של האוטומט.

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

עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים,

ולסמן את המצב המקבל / המצבים המקבלים.

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

16. נבנתה מכונת טיורינג המחשבת את הפונקציה $f(m, n)$. מכונה זו מוצאת את המספר הקטן מבין שני מספרים m ו- n . המכונה מקבלת כקלט שני מספרים m ו- n שלמים וגדולים מ-0. שני המספרים רשומים על הסרט כמספרים אונריים (מספר אונרי m הוא מספר המיוצג על ידי m תווים של 1) ובין שני המספרים רשום הסימן #. הפלט של המכונה הוא המספר הקטן מבין שני המספרים, והוא יירשם על הסרט כמספר אונרי בין שני סימני \$. סימן ה-\$ השמאלי יירשם במקום הסימן #, והמספר יירשם מימינו. במהלך הפעולה המכונה יכולה להיעזר בסימנים a, b . לדוגמה: בעבור הקלט $m = 3$ ו- $n = 5$, לפני החישוב סרט הזיכרון ייראה כך:

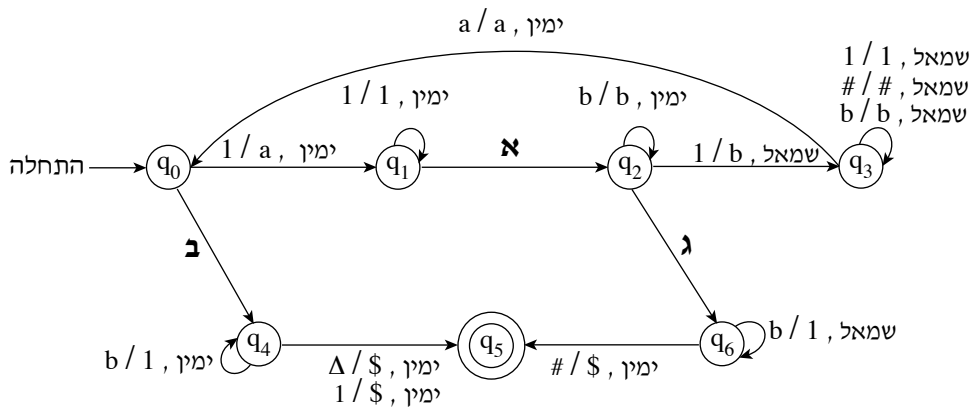
⊢	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	...
---	---	---	---	---	---	---	---	---	---	---	---	---	-----

לאחר סיום החישוב ייראה הסרט כך:

⊢	\$	1	1	1	\$...
---	-----	-----	-----	----	---	---	---	----	-----

הערה: התאים המסומנים ב-... הם תאים שאין חשיבות לתוכנם.

לפניך סרטוט חלקי של המכונה.



א. בסרטוט יש שלושה מעברים המסומנים באותיות א-ג.

במעברים אלה חסרים סימני הקלט וההוראות.

העתק למחברתך את הסרטוט, והשלם את שלושת המעברים החסרים כך שהמכונה

תחשב את הפונקציה $f(m, n)$.

ב. הראה את תהליך החישוב של המכונה בעבור הקלט $m = 1, n = 1$.

בכל שלב רשום את מצב הסרט, סמן היכן נמצא ראש המכונה, ורשום באיזה

מהמצבים $q_0 - q_6$ המכונה נמצאת.

תכנות מונחה עצמים

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

17. בגן החיות "ZooZoo" בעלי החיים יודעים "לדבר". חיות רעבות משמיעות את קולן בדרישה לקבל אוכל. כאשר מטפל מגיע להאכיל את החיות, החיות מקבלות את פניו בקריאות שמחה. החיות משמיעות קול כשהן מסיימות לאכול.
לפניך פרויקט המדמה את גן החיות (ZooZoo), את בעלי החיים בגן החיות ואת "אמרות החכמה" שהם משמיעים.
בעל חיים (Animal) יכול להיות זוחל (Reptile) או בעל חיים ימי (Marine).
זוחל (Reptile) יכול להיות נחש (Snake) או תנין (Crocodile),
בעל חיים ימי (Marine) יכול להיות דג זהב (GoldFish).
לכל בעל חיים נשמר שמו (name). בעבור זוחל נשמר גם האורך שלו (len),
ובעבור בעל חיים ימי נשמר גם עומק הצלילה המקסימלי שלו (depth).
הפעולה hungry מחזירה את דבריו של בעל חיים רעב.
הפעולה caretaker מחזירה את דבריו של בעל חיים המקבל את פני המטפל שלו.
הפעולה satisfied מחזירה את דבריו של בעל חיים שסיים לאכול.

```
public class Animal
{
    private String name;           // שם בעל החיים
    public static int count = 0;   // מונה בעלי חיים

    public Animal(String name)
    {
        this.name = name;
        count++;
    }
    public String hungry() { return this.name + " is hungry!"; } // רעב
    public String caretaker() { return "Yammi"; } // קבלת המטפלים
    public String satisfied() { return "Finish eating"; } // סיים לאכול
}
public class Reptile extends Animal
{
    private int len;              // אורך
    public Reptile(String name, int len)
    {
        super(name);
        this.len = len;
    }
    public String caretaker() { return "Crawling: " + super.caretaker(); }
}
(שים לב: המשך השאלה בעמוד הבא.)
/המשך בעמוד 34/
```

```

public class Snake extends Reptile
{
    public Snake(String name, int len) { super(name, len); }
    public String satisfied() { return "Tssss"; }
}

public class Crocodile extends Reptile
{
    public Crocodile(int len) { super("Crocki" + count , len); }
    public String satisfied() { return super.satisfied() + " Whaamm"; }
}

public class Marine extends Animal
{
    private double depth; // עומק

    public Marine(String name, double depth)
    {
        super (name);
        this.depth = depth;
    }
    public String caretaker() { return "Swimming"; }
}

public class GoldFish extends Marine
{
    public GoldFish(String name) { super(name, 0.2); }
    public String satisfied() { return "Bloop bloop"; }
}

```

```

public class ZooZoo
{
    private Animal[] animals;



    public ZooZoo()
    {
        this.animals = new Animal[4];
        this.animals[0] = new Snake("snaki", 50);
        this.animals[1] = new Crocodile(78);
        this.animals[2] = new GoldFish("goldi");
        this.animals[3] = new Crocodile(103);
    }

    public void print()
    {
        for (int i = 0; i < this.animals.length; i++)
        {
            System.out.println(this.animals[i].hungry());
            System.out.println(this.animals[i].caretaker());
            System.out.println(this.animals[i].satisfied());
            System.out.println("*****");
        }
    }
}

public class Test
{
    public static void main(String [] args)
    {
        ZooZoo zoo = new ZooZoo();
        zoo.print();
    }
}

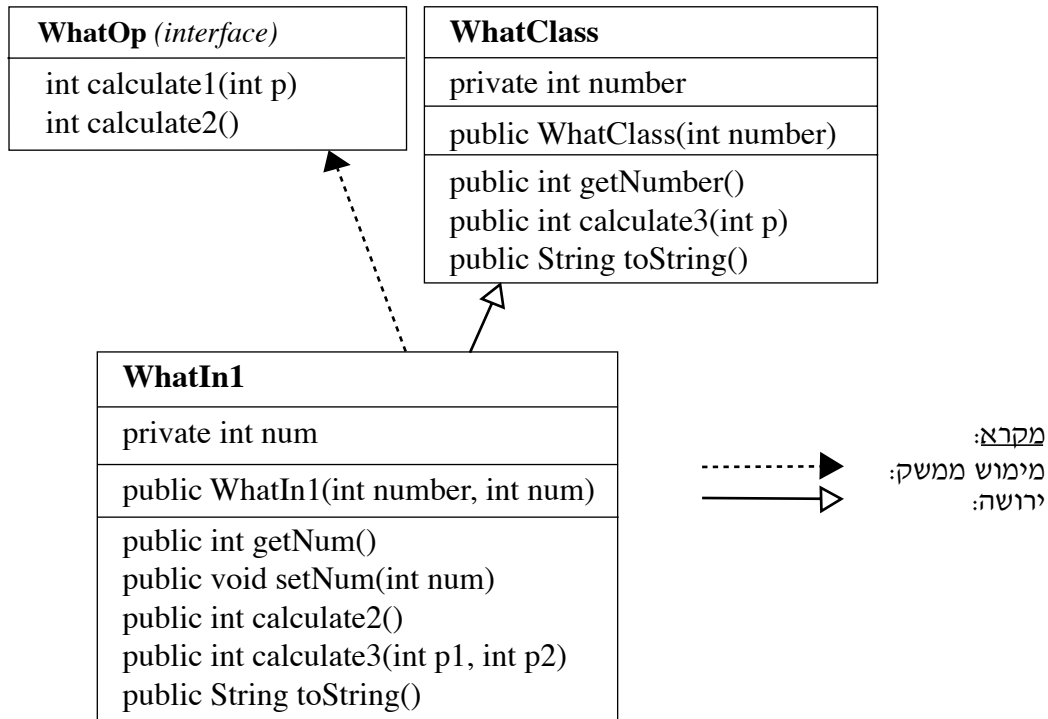
```

א. סרטט את היררכיית המחלקות המוגדרות בפרויקט. השתמש בסימנים האלה:

 ירשה
 הכלה

ב. כתוב מעקב אחר הפעולה main שבמחלקה Test, וכתוב את הפלט.
 במעקב יש לכלול את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

18. לפניך UML חלקי:



- א. האם מימוש הפעולה `public int calculate1(int p)` במחלקה **WhatClass**, יענה על דרישת המימוש שלה במחלקה **WhatIn1**? נמק את תשובתך.
- ב. מימוש הפעולה הבונה במחלקה **WhatIn1** הוא:

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
    
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ג. במחלקה **WhatIn1** מומשה הפעולה `calculate2()`, המחזירה את הערך השלם של ממוצע תכונות העצם:

```
public int calculate2()
{   return (int)((this.number + this.num)/2); }
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ד. לפניך תיאור חלקי של המחלקה **WhatIn2**:

class WhatIn2 extends WhatIn1	
private int sum	
public WhatIn2 (int number, int num, int sum)	
public int calculate3(int p1, int p2, int p3)	

(1) האם אפשר להסתמך על הפעולה הבונה ברירת מחדל במקום להגדיר פעולה בונה במחלקה **WhatIn2**? נמק את תשובתך.

(2) הפעולה `calculate3` מומשה בשלוש המחלקות **WhatIn1**, **WhatClass** ו- **WhatIn2** באופן הזה:

class WhatClass	public int calculate3(int p) { return this.number * p; }
class WhatIn1	public int calculate3(int p1, int p2) { return this.calculate3(p1) + this.num * p2 * p2; }
class WhatIn2	public int calculate3(int p1, int p2, int p3) { return this.calculate3(p1, p2) + this.sum * p3 * p3 * p3; }

הנח כי העצם `obj` הוא מטיפוס **WhatIn2**, וערכי התכונות שלו הם:

number = 1

num = 2

sum = 3

לפניך הוראה הכתובה בפעולה הראשית:

```
System.out.println(obj.calculate3(1000, 100, 10));
```

הראה את המעקב אחר ביצוע ההוראה. במעקב התייחס לזימון פעולות ולערכי תכונות העצם. רשום את הפלט המתקבל.

19. לפניך כמה עקרונות של תכנות מונחה עצמים:

- הכמסה – encapsulation
 - העמסה – overloading
 - ירושה – inheritance
 - דריסה – overriding
 - רב-צורתיות – polymorphism
- בשאלה זו תהיה התייחסות לחלק מהם.

לפניך מימוש חלקי של שתי מחלקות: **Stam, Davar**.

```
public class Stam
{
    private char x;

    public Stam()        { this.x = '*'; }
    public Stam(char c)  { this.x = c; }
    public Stam getStam() { return this; }
    public String toString() { return "x=" + this.x; }
    public boolean isSame1(Stam other)
    { // התייחסות למימוש בהמשך השאלה }
    public boolean isSame2(Stam other)
    { // התייחסות למימוש בהמשך השאלה }
    public void same(Stam other)
    {
        if (this.isSame1(other))
            System.out.println(this + " same1 as " + other);
        else
            System.out.println(this + " not same1 as " + other);
        if (this.isSame2(other))
            System.out.println(this + " same2 as " + other);
        else
            System.out.println(this + " not same2 as " + other);
    }
    public void print()        { System.out.println(this.toString ()); }
    public void print(Stam other) { this.same(other); }
}

```

/המשך בעמוד 39/

```
public class Davar extends Stam
{
    private int y;
    public Davar()                { super(); this.y = 0; }
    public Davar(char c)          { super(c); this.y = 0; }
    public Davar(char c , int num) { super(c); this.y = num; }
    public String toString()      { return "Davar: " + super.toString(); }
}
```

א. במחלקה **Stam** הוגדרו שתי פעולות בונות.

(1) מהו העיקרון של תכנות מונחה עצמים (מבין העקרונות שצוינו בתחילת

השאלה) המאפשר זאת?

(2) כיצד בוחר המהדר (קומפיילר) איזו פעולה בונה להפעיל?

ב. איזה עיקרון של תכנות מונחה עצמים (מבין העקרונות שצוינו בתחילת השאלה)

ממומש בפעולה `toString()` במחלקה **Davar** ? נמק.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. נתונה המחלקה הראשית שלהלן:

```
public class Program
{
    public static void main(String[] args)
    {
        Stam[] s = new Stam[6];
        s[0] = new Stam();
        s[1] = new Davar();
        s[2] = new Stam('b');
        s[3] = new Davar('b');
        s[4] = new Davar('a', 0);
        s[5] = s[2].getStam();

        for (int i = 0; i < s.length; i++)
        {
            s[i].print();
        }

        s[1].print(s[0]);
        s[2].print(s[5]);
        s[3].print(s[4]);
    }
}
```

- (1) הצג את המערך s הנבנה בקטע 1 (במחלקה הראשית הנתונה).
בעבור כל אחד מהעצמים רשום את ערכי התכונות שלו.
- (2) רשום את פלט הלולאה בקטע 2.
- (3) איזה עיקרון של תכנות מונחה עצמים (מבין העקרונות שצוינו בתחילת השאלה) בא לידי ביטוי בקטע 2?
- (4) ממש את הפעולות isSame1(), isSame2() שבמחלקה **Stam** כך שהפלט המתקבל מקטע 3, יהיה הפלט שלהלן:

```
Davar: x = * same1 as x = *
Davar: x = * not same2 as x = *
x = b same1 as x = b
x = b same2 as x = b
Davar: x = b not same1 as Davar: x = a
Davar: x = b not same2 as Davar: x = a
```


20. במפעל לייצור נרות יש מקום ל-12 פסי ייצור לכל היותר. בכל פס ייצור מייצרים דגם אחד של נרות. לכל דגם יש כמה צבעים שאפשר לייצר מהם את הנרות מאותו דגם. כל נר הוא בצבע אחד בלבד מבין הצבעים בדגם שלו.
- בעבור כל דגם של נרות נשמר המידע: שם הדגם, קוד הדגם, שמות הצבעים שמהם הוא מיוצר, כמות הנרות שיש במלאי מכל אחד מהצבעים של הדגם, וכמות הנרות המקסימלית מדגם זה שאפשר לאחסן (מכל הצבעים יחד).
- דוגמה לדגם נרות:
- שם: "שושנה".
 - קוד: 11.
 - שמות הצבעים שמהם הוא מיוצר: אדום, לבן, צהוב.
 - כמות הנרות במלאי: 200 נרות אדומים, 350 נרות לבנים ו-70 נרות צהובים.
 - הכמות המקסימלית של הנרות מדגם זה (מכל הצבעים יחד) שאפשר לאחסן: 100,000.

- במפעל נשמר מידע בנוגע לכל דגמי הנרות המיוצרים בו.
- בכל פעם שרוצים לייצר נר מדגם מסוים יש להתייחס לתנאים האלה:
- צבע הנר ייבחר על פי כמות הנרות במלאי: הצבע שכמות הנרות ממנו במלאי היא הקטנה ביותר מאותו דגם, הוא הצבע שייבחר לייצור הנר (אם יש יותר מצבע אחד כזה ייבחר אחד מהם).
 - כמות הייצור נתונה לבחירת המפעיל, אך אסור שכמות הנרות הכוללת מדגם מסוים (הנרות שייוצרו והנרות שבמלאי) תעלה על הכמות המקסימלית מדגם זה שאפשר לאחסן.

מהנדס התוכנה של המפעל בנה פרויקט ובו שלוש מחלקות:

CandleKind	מחלקה המייצגת דגם של נרות
Factory	מחלקה המייצגת את המפעל
Run	מחלקה ראשית

(שים לב: המשך השאלה בעמוד הבא.)

לפניך חלקים מן המחלקה **CandleKind**:

public class CandleKind	
private String name;	// שם הדגם
private int code;	// קוד הדגם
private String[] colors;	// מערך של מגוון צבעי הדגם
private int[] amounts;	// כמות במלאי מכל צבע (בהתאמה)
private int maxTotalAmount;	// מכסה מקסימלית לאחסון
public CandleKind(String name, int code, String[] colors, int maxTotalAmount) { this.name = name; this.code = code; this.colors = colors; this.amounts = new int[colors.length]; for (int i = 0; i < amounts.length; i++) { this.amounts[i] = 0; } this.maxTotalAmount = maxTotalAmount; }	
public void update(String color, int amount) // הפעולה מגדילה בכמות amount את כמות הנרות בצבע color שבמלאי	

- א.** כתוב הוראה / הוראות לבניית עצם בשם kind1 מן המחלקה **CandleKind** בעבור דגם חדש של נרות שמתחילים לייצר במפעל. הדגם החדש מיוצר בשלושה צבעים שונים. בחר ערכים לתכונות כרצונך. רשום את ערכי תכונות העצם לאחר בנייתו.
- ב.** כתוב את כותרת המחלקה **Factory** ואת התכונה / התכונות שלה. כתוב תיעוד לכל תכונה.
- ג.** לפניך טבלה עם תיעוד של פעולות 1-3 שיש להוסיף לפרויקט. לכל פעולה כתוב:
- i** באיזו מחלקה (מבין שלוש המחלקות) מתאים להגדיר אותה. הסבר את בחירתך.
 - ii** את הכותרת של הפעולה.

1	פעולה המחזירה צבע של נר מדגם נרות מסוים שיתחילו לייצר.
2	פעולה המחזירה את כמות הנרות שאפשר לייצר מדגם נרות מסוים.
3	פעולה המחזירה את הקוד של דגם הנר שבעבורו ההפרש בין הכמות שיש מדגם זה במלאי לבין הכמות שאפשר לאחסן הוא הגדול ביותר. אם יש כמה דגמי נרות כאלה, יוחזר אחד מהם. אם הכמות במלאי של כל אחד מדגמי הנרות היא המקסימלית שאפשר לאחסן, יוחזר הקוד 999 (קוד לא קיים).

ד. לפניך פעולה ראשית הממומשת במחלקה **Run**. הפעולה מנהלת את תהליך הייצור. השלם את קטעי הקוד הממוספרים (1)-(4) על פי פירוט הדרישות המופיע אחרי הקוד. הנח כי הפעולות שהגדרת בסעיף ג קיימות, ומוגדרות פעולות `get`, `set` לכל תכונה במחלקה **CandleKind** הנתונה ובמחלקה **Factory** שכתבת בסעיף ב. אם אתה מגדיר פעולות נוספות, רשום באיזו מחלקה הן מוגדרות וממש אותן באופן מלא.

```
public static void main(String [] args)
{
    Factory fty;
    // מכאן והלאה הנח שקיים העצם fty עם ערכי
    // תכונות המתארים את מצב המלאי במפעל
    int code = _____ (1)

    while (code != 999)
    {
        CandleKind ck;
        _____ (2)
        String colorToProduce;
        _____ (3)
        int amount;
        _____ (4)
        ck.update(colorToProduce, amount);
        code = _____ (1)
    }
}
```

פירוט הדרישות:

- (1) זימון פעולה 3 מהטבלה שבסעיף ג.
- (2) קטע קוד אשר יביא לכך שהמשתנה `ck` יכיל את דגם הנר בעל הקוד `code`.
- (3) קטע קוד אשר יביא לכך שהמשתנה `colorToProduce` יכיל את הצבע שיש לייצר מדגם הנר שהקוד שלו הוא `code`.
- (4) קטע קוד אשר יביא לכך שהמשתנה `amount` יכיל את כמות הנרות שאפשר לייצר מדגם הנר שהקוד שלו הוא `code`.

שים לב: תכנות מונחה עצמים ב-C# מתחיל בעמוד 45

תכנות מונחה עצמים

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

21. בגן החיות "ZooZoo" בעלי החיים יודעים "לדבר". חיות רעבות משמיעות את קולן בדרישה לקבל אוכל. כאשר מטפל מגיע להאכיל את החיות, החיות מקבלות את פניו בקריאות שמחה. החיות משמיעות קול כשהן מסיימות לאכול. לפניך פרויקט המדמה את גן החיות (**ZooZoo**), את בעלי החיים בגן החיות ואת "אמרות החכמה" שהם משמיעים. בעל חיים (**Animal**) יכול להיות זוחל (**Reptile**) או בעל חיים ימי (**Marine**). זוחל (**Reptile**) יכול להיות נחש (**Snake**) או תנין (**Crocodile**), בעל חיים ימי (**Marine**) יכול להיות דג זהב (**GoldFish**). לכל בעל חיים נשמר שמו (**name**). בעבור זוחל נשמר גם האורך שלו (**len**), ובעבור בעל חיים ימי נשמר גם עומק הצלילה המקסימלי שלו (**depth**). הפעולה Hungry מחזירה את דבריו של בעל חיים רעב. הפעולה Caretaker מחזירה את דבריו של בעל חיים המקבל את פני המטפל שלו. הפעולה Satisfied מחזירה את דבריו של בעל חיים שסיים לאכול.

```
public class Animal
{
    private string name;           // שם בעל החיים
    public static int count = 0;   // מונה בעלי חיים

    public Animal(string name)
    {
        this.name = name;
        count++;
    }

    public string Hungry() { return this.name + " is hungry!"; } // רעב
    public virtual string Caretaker() { return "Yammi"; } // קבלת המטפלים
    public virtual string Satisfied() { return "Finish eating"; } // סיים לאכול
}

public class Reptile: Animal
{
    private int len;              // אורך

    public Reptile(string name, int len): base(name)
    {
        this.len = len;
    }

    public override string Caretaker() {return "Crawling: " + base.Caretaker(); }
}
/המשך בעמוד 46/ (שים לב: המשך השאלה בעמוד הבא.)
```

```

public class Snake: Reptile
{
    public Snake(string name, int len): base (name, len) { }
    public override string Satisfied() { return "Tssss"; }
}

public class Crocodile: Reptile
{
    public Crocodile(int len): base("crocki" + count , len){ }
    public override string Satisfied() { return base.Satisfied() + " Whaamm"; }
}

public class Marine: Animal
{
    private double depth;           // עומק

    public Marine(string name, double depth): base(name)
    {
        this.depth = depth;
    }
    public override string Caretaker() { return "Swimming"; }
}

public class GoldFish: Marine
{
    public GoldFish(string name): base(name, 0.2) { }
    public override string Satisfied() { return "Bloop bloop"; }
}

```

```

public class ZooZoo
{
    private Animal[] animals;

    public ZooZoo()
    {
        this.animals = new Animal[4];
        this.animals[0] = new Snake("snaki", 50);
        this.animals[1] = new Crocodile(78);
        this.animals[2] = new GoldFish("goldi");
        this.animals[3] = new Crocodile(103);
    }

    public void Print()
    {
        for (int i = 0; i < this.animals.Length; i++)
        {
            Console.WriteLine(this.animals[i].Hungry());
            Console.WriteLine(this.animals[i].Caretaker());
            Console.WriteLine(this.animals[i].Satisfied());
            Console.WriteLine("****");
        }
    }
}

public class Test
{
    public static void Main(string [] args)
    {
        ZooZoo zoo = new ZooZoo();
        zoo.Print();
    }
}

```

א. סרטט את היררכיית המחלקות המוגדרות בפרויקט. השתמש בסימנים האלה:

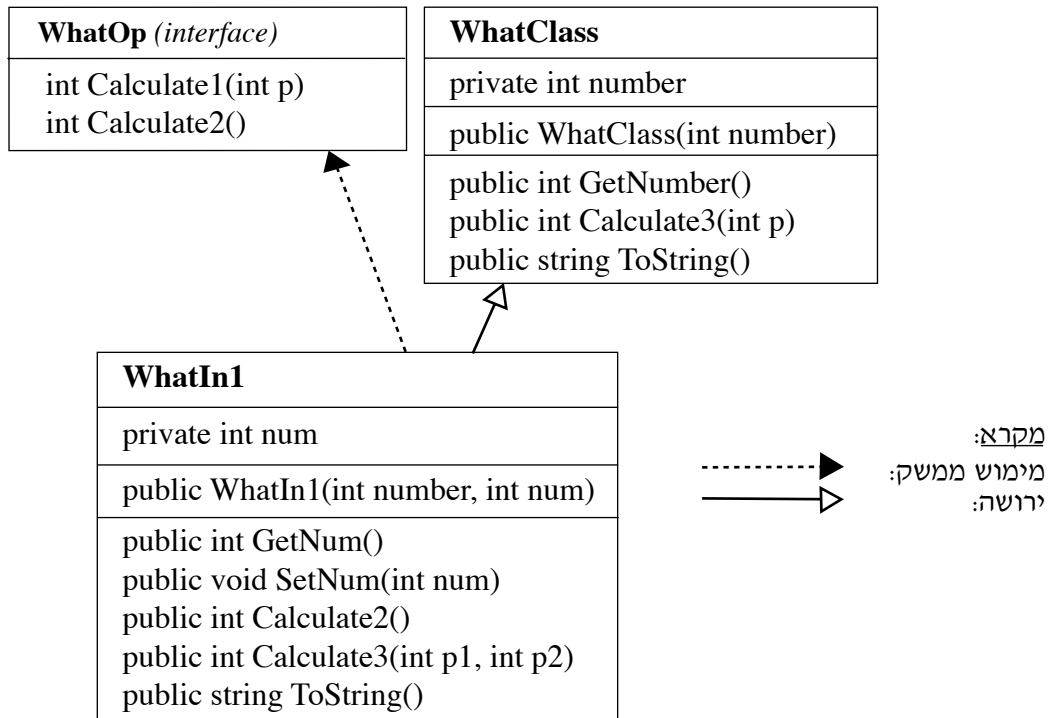
—▷ ירושה

—◆ הכלה

ב. כתוב מעקב אחר הפעולה Main שבמחלקה Test, וכתוב את הפלט.

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

22. לפניך UML חלקי:



א. האם מימוש הפעולה `public int Calculate1(int p)` במחלקה **WhatClass**, יענה על דרישת המימוש שלה במחלקה **WhatIn1**? נמק את תשובתך.

ב. מימוש הפעולה הבונה במחלקה **WhatIn1** הוא:

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
    
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ג. במחלקה **WhatIn1** מומשה הפעולה `Calculate2()`, המחזירה את הערך השלם של ממוצע תכונות העצם:

```
public int Calculate2()
{ return (int)((this.number + this.num)/2); }
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ד. לפניך תיאור חלקי של המחלקה **WhatIn2**:

<code>class WhatIn2: WhatIn1</code>
<code>private int sum</code>
<code>public WhatIn2(int number, int num, int sum)</code>
<code>public int Calculate3(int p1, int p2, int p3)</code>

(1) האם אפשר להסתמך על הפעולה הבונה ברירת מחדל במקום להגדיר פעולה בונה במחלקה **WhatIn2**? נמק את תשובתך.

(2) הפעולה `Calculate3` מומשה בשלוש המחלקות **WhatIn1**, **WhatClass** ו- **WhatIn2** באופן הזה:

<code>class WhatClass</code>	<code>public int Calculate3(int p)</code> <code>{ return this.number * p; }</code>
<code>class WhatIn1</code>	<code>public int Calculate3 (int p1, int p2)</code> <code>{ return this.Calculate3(p1) + this.num * p2 * p2; }</code>
<code>class WhatIn2</code>	<code>public int Calculate3 (int p1, int p2, int p3)</code> <code>{ return this.Calculate3(p1, p2) + this.sum * p3 * p3 * p3;}</code>

הנח כי העצם `obj` הוא מטיפוס **WhatIn2**, וערכי התכונות שלו הם:

`number = 1`

`num = 2`

`sum = 3`

לפניך הוראה הכתובה בפעולה הראשית:

```
Console.WriteLine(obj.Calculate3(1000, 100, 10));
```

הראה את המעקב אחר ביצוע ההוראה. במעקב התייחס לזימון פעולות ולערכי תכונות העצם. רשום את הפלט המתקבל.

23. לפניך כמה עקרונות של תכנות מונחה עצמים:

- הכמסה – encapsulation
- העמסה – overloading
- ירושה – inheritance
- דריסה – overriding
- רב-צורתיות – polymorphism

בשאלה זו תהיה התייחסות לחלק מהם.

לפניך מימוש חלקי של שתי מחלקות: **Stam, Davar**.

```
public class Stam
{
    private char x;

    public Stam()           { this.x = '*'; }
    public Stam(char c)    { this.x = c; }
    public Stam GetStam()  { return this; }
    public override string ToString() { return "x=" + this.x; }
    public bool IsSame1(Stam other)
    { // התייחסות למימוש בהמשך השאלה }
    public bool IsSame2(Stam other)
    { // התייחסות למימוש בהמשך השאלה }
    public void Same(Stam other)
    {
        if (this.IsSame1(other))
            Console.WriteLine(this + " same1 as " + other);
        else
            Console.WriteLine(this + " not same1 as " + other);
        if (this.IsSame2(other))
            Console.WriteLine(this + " same2 as " + other);
        else
            Console.WriteLine(this + " not same2 as " + other);
    }
    public void Print()     { Console.WriteLine(this.ToString ()); }
    public void Print(Stam other) { this.Same(other); }
}
}
```

/המשך בעמוד 51/

```
public class Davar: Stam
{
    private int y;
    public Davar(): base()           { this.y = 0; }
    public Davar(char c) : base(c)   { this.y = 0; }
    public Davar(char c , int num): base(c) { this.y = num; }
    public override string ToString() { return "Davar: " + base.ToString(); }
}
```

א. במחלקה **Stam** הוגדרו שתי פעולות בונות.

(1) מהו העיקרון של תכנות מונחה עצמים (מבין העקרונות שצוינו בתחילת השאלה)

המאפשר זאת?

(2) כיצד בוחר המהדר (קומפיילר) איזו פעולה בונה להפעיל?

ב. איזה עיקרון של תכנות מונחה עצמים (מבין העקרונות שצוינו בתחילת השאלה)

ממומש בפעולה ToString() במחלקה **Davar** ? נמק.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. נתונה המחלקה הראשית שלהלן:

```
public class Program
{
    public static void Main(string[] args)
    {
        Stam[] s = new Stam[6];
        s[0] = new Stam();
        s[1] = new Davar();
        s[2] = new Stam('b');
        s[3] = new Davar('b');
        s[4] = new Davar('a', 0);
        s[5] = s[2].GetStam();

        for (int i = 0; i < s.Length; i++)
        {
            s[i].Print();
        }

        s[1].Print(s[0]);
        s[2].Print(s[5]);
        s[3].Print(s[4]);
    }
}
```

- (1) הצג את המערך s הנבנה בקטע 1 (במחלקה הראשית הנתונה).
בעבור כל אחד מהעצמים רשום את ערכי התכונות שלו.
- (2) רשום את פלט הלולאה בקטע 2.
- (3) איזה עיקרון של תכנות מונחה עצמים (מבין העקרונות שצוינו בתחילת השאלה) בא לידי ביטוי בקטע 2?
- (4) ממש את הפעולות IsSame1() , IsSame2() שבמחלקה **Stam** כך שהפלט המתקבל מקטע 3, יהיה הפלט שלהלן:

```
Davar: x = * same1 as x = *
Davar: x = * not same2 as x = *
x = b same1 as x = b
x = b same2 as x = b
Davar: x = b not same1 as Davar: x = a
Davar: x = b not same2 as Davar: x = a
```

24. במפעל לייצור נרות יש מקום ל- 12 פסי ייצור לכל היותר. בכל פס ייצור מייצרים דגם אחד

של נרות. לכל דגם יש כמה צבעים שאפשר לייצר מהם את הנרות מאותו הדגם. כל נר הוא בצבע אחד בלבד מבין הצבעים בדגם שלו.

בעבור כל דגם של נרות נשמר המידע: שם הדגם, קוד הדגם, שמות הצבעים שמהם הוא מיוצר, כמות הנרות שיש במלאי מכל אחד מהצבעים של הדגם, וכמות הנרות המקסימלית מדגם זה שאפשר לאחסן (מכל הצבעים יחד).

דוגמה לדגם נרות:

— שם: "שושנה".

— קוד: 11.

— שמות הצבעים שמהם הוא מיוצר: אדום, לבן, צהוב.

— כמות הנרות במלאי: 200 נרות אדומים, 350 נרות לבנים ו-70 נרות צהובים.

— הכמות המקסימלית של הנרות מדגם זה (מכל הצבעים יחד) שאפשר לאחסן:

100,000.

במפעל נשמר מידע בנוגע לכל דגמי הנרות המיוצרים בו.

בכל פעם שרוצים לייצר נר מדגם מסוים יש להתייחס לתנאים האלה:

- צבע הנר ייבחר על פי כמות הנרות במלאי: הצבע שכמות הנרות ממנו במלאי היא הקטנה ביותר מאותו דגם, הוא הצבע שייבחר לייצור הנר (אם יש יותר מצבע אחד כזה ייבחר אחד מהם).
- כמות הייצור נתונה לבחירת המפעיל, אך אסור שכמות הנרות הכוללת מדגם מסוים (הנרות שייצרו והנרות שבמלאי) תעלה על הכמות המקסימלית מדגם זה שאפשר לאחסן.

מהנדס התוכנה של המפעל בנה פרויקט ובו שלוש מחלקות:

CandleKind	מחלקה המייצגת דגם של נרות
Factory	מחלקה המייצגת את המפעל
Run	מחלקה ראשית

(שים לב: המשך השאלה בעמוד הבא.)

לפניך חלקים מן המחלקה **CandleKind**:

public class CandleKind	
private string name;	// שם הדגם
private int code;	// קוד הדגם
private string[] colors;	// מערך של מגוון צבעי הדגם
private int[] amounts;	// כמות במלאי מכל צבע (בהתאמה)
private int maxTotalAmount;	// מכסה מקסימלית לאחסון
public CandleKind(string name, int code, string[] colors, int maxTotalAmount) { this.name = name; this.code = code; this.colors = colors; this.amounts = new int[colors.Length]; for (int i = 0; i < amounts.Length; i++) { this.amounts[i] = 0; } this.maxTotalAmount = maxTotalAmount; }	
public void Update(string color, int amount) // הפעולה מגדילה בכמות amount את כמות הנרות בצבע color שבמלאי	

- א.** כתוב הוראה / הוראות לבניית עצם בשם kind1 מן המחלקה **CandleKind** בעבור דגם חדש של נרות שמתחילים לייצר במפעל. הדגם החדש מיוצר בשלושה צבעים שונים. בחר ערכים לתכונות כרצונך. רשום את ערכי תכונות העצם לאחר בנייתו.
- ב.** כתוב את כותרת המחלקה **Factory** ואת התכונה / התכונות שלה. כתוב תיעוד לכל תכונה.
- ג.** לפניך טבלה עם תיעוד של פעולות 1-3 שיש להוסיף לפרויקט. לכל פעולה כתוב:
- i** באיזו מחלקה (מבין שלוש המחלקות) מתאים להגדיר אותה. הסבר את בחירתך.
 - ii** את הכותרות של הפעולה.

1	פעולה המחזירה צבע של נר מדגם נרות מסוים שיתחילו לייצר.
2	פעולה המחזירה את כמות הנרות שאפשר לייצר מדגם נרות מסוים.
3	פעולה המחזירה את הקוד של דגם הנר שבעבורו ההפרש בין הכמות שיש מדגם זה במלאי לבין הכמות שאפשר לאחסן הוא הגדול ביותר. אם יש כמה דגמי נרות כאלה, יוחזר אחד מהם. אם הכמות במלאי של כל אחד מדגמי הנרות היא המקסימלית שאפשר לאחסן, יוחזר הקוד 999 (קוד לא קיים).

ד. לפניך פעולה ראשית הממומשת במחלקה **Run**. הפעולה מנהלת את תהליך הייצור. השלם את קטעי הקוד הממוספרים (1)-(4) על פי פירוט הדרישות המופיע אחרי הקוד. הנח כי הפעולות שהגדרת בסעיף ג קיימות, ומוגדרות פעולות **Get**, **Set** לכל תכונה במחלקה **CandleKind** הנתונה ובמחלקה **Factory** שכתבת בסעיף ב. אם אתה מגדיר פעולות נוספות, רשום באיזו מחלקה הן מוגדרות וממש אותן באופן מלא.

```
public static void Main(string [] args)
{
    Factory fty;
    //          מכאן והלאה הנח שקיים העצם fty עם ערכי
    //          תכונות המתארים את מצב המלאי הקיים במפעל
    int code = _____ (1)

    while (code != 999)
    {
        CandleKind ck;
        _____ (2)

        string colorToProduce;
        _____ (3)

        int amount;
        _____ (4)

        ck.Update(colorToProduce, amount);
        code = _____ (1)
    }
}
```

פירוט הדרישות:

- (1) זימון פעולה 3 מהטבלה שבסעיף ג.
- (2) קטע קוד אשר יביא לכך שהמשתנה ck יכיל את דגם הנר בעל הקוד code.
- (3) קטע קוד אשר יביא לכך שהמשתנה colorToProduce יכיל את הצבע שיש לייצר מדגם הנר שהקוד שלו הוא code.
- (4) קטע קוד אשר יביא לכך שהמשתנה amount יכיל את כמות הנרות שאפשר לייצר מדגם הנר שהקוד שלו הוא code.

בהצלחה!