

## תבניות – פרק 7

### מנייה וצבירה

נתבונן בשתי הבעיות האלגוריתמיות הבאות:

**בעיה 1:** עליה על מתקן "רכבת הרים" מותרת רק לילדים שגובהם עולה על 1.40 מטר וגילם עולה על 8. כתבו אלגוריתם שהקלט שלו 40 זוגות ערכים המייצגים את הגובה והגיל של 40 ילדים, והפלט שלו הוא מספר הילדים הרשאים לעלות למתקן.

**בעיה 2:** מנהל האתר הבית-ספרי "ישראלנד" מעוניין לדעת מהו מספר הכניסות לאתר ביממה. כתבו אלגוריתם שהקלט שלו הוא סדרה של קודי משתמשים, המסתיימת בזקוף 0, והפלט שלו הוא מספר הכניסות לאתר ביממה.

בשתי הבעיות האלגוריתמיות עלינו **למנות** (לספור) ערכים. בבעיה 1 יש למנות את מספר הילדים הרשאים לעלות למתקן "רכבת הרים" ובבעיה 2 יש למנות את מספר הכניסות לאתר ביממה. מנייה היא אחת התבניות השימושיות ביותר בפיתוח אלגוריתמים. השימוש בתבנית זו נעשה בכל פעם שיש למנות כמות הופעות של ערך.

נתבונן בשני האלגוריתמים הבאים, הפותרים את שתי הבעיות שלעיל:

1. אגא אג count 0-2	1. אגא אג count 0-2
2. קאוט קוז ממת code-2	2. כצ 40 פמניס:
3. כא עוז 0 < code כצ:	2.1. קאוט גובה כ height-2 גזי age-2
3.1. הכז אג count 1-2	2.2. אס > 1.4 אס > 8 age
3.2. קאוט קוז ממת code-2	2.2.1. אג count 1-2
4. הכז כפוט אג ערכו אג count	3. הכז כפוט אג ערכו אג count

בשני האלגוריתמים ערכו של המונה (count) מאותחל לפני הלולאה ל-0 ובגוף הלולאה גדל ערכו של המונה ב-1. שימו לב כי בפתרון לבעיה 1 מספר הפעמים שמבוצע גוף הלולאה ידוע מראש (40) ואילו בפתרון לבעיה 2 מספר הפעמים שמבוצע גוף הלולאה אינו ידוע מראש (סדרת נתוני הקלט מסתיימת עם קליטת הזקוף 0).

עתה נתבונן בשתי הבעיות האלגוריתמיות הבאות :

**בעיה 3 :** כתבו אלגוריתם שהקלט שלו הוא 15 מספרים ממשיים והפלט שלו הוא סכום המספרים החיוביים.

**בעיה 4 :** בתחרות הרמת משקולות משקלה של המשקולת ההתחלתית הוא 20 ק"ג. כל המתחרים בתחרות מסוגלים להרים משקולת זו. עם התקדמות התחרות נוספות עוד ועוד משקולות למשקולת ההתחלתית. כתבו אלגוריתם שהקלט שלו הוא סדרה של זוגות נתונים : הנתון הראשון הוא משקל המשקולת אותה מוסיפים למשקל המצטבר שהרים מתחרה, והנתון השני הוא התו 'y' אם המתחרה הצליח להרים את המשקל המצטבר כולל המשקולת החדשה, והתו 'n' אם לא הצליח. התו 'n' יופיע רק בזוג הנתונים האחרון. הפלט של האלגוריתם הוא המשקל המצטבר של המשקולות שהמתחרה הצליח להרים.

בשתי הבעיות האלגוריתמיות עלינו **לצבור** (בדוגמה זו, לחשב סכום) ערכים. בבעיה 3 יש לצבור מספרים חיוביים ובבעיה השנייה יש לצבור את משקלי המשקולות. תבנית הצבירה, בדומה לתבנית מנייה, הינה תבנית שימושית ביותר בפיתוח אלגוריתמים. השימוש בתבנית זו נעשה בכל פעם שיש לצבור סדרת ערכים בצורה כלשהי, למשל על ידי סכום או על ידי מכפלה.

נתבונן בשני האלגוריתמים, הפותרים את שתי הבעיות שלעיל :

<p>1. אגוא אג sum 20-2</p> <p>2. קאוט משקל משקולת 2-weight ונ המשק 2-continuation</p> <p>3. כו ע/ו 'n' &lt; continuation ככ3:</p> <p>3.1. הוול 1-sum אג weight</p> <p>3.2. קאוט משקל משקולת 2-weight ונ continuation-2 המשק</p> <p>4. ה3ג כפלט אג ערכו ו sum</p>	<p>1. אגוא אג sum 0-2</p> <p>2. ככ3 15 עעמיו:</p> <p>2.1. קאוט מספר מנשי 2-num</p> <p>2.2. אג &gt; 0 num</p> <p>2.1.1. הוול 1-sum אג num</p> <p>3. ה3ג כפלט אג ערכו ו sum</p>
--	---

בשני האלגוריתמים ערכו של הצובר (sum) מאותחל לפני הלולאה לערך תחילי, ולהבדיל ממונה, ערך זה אינו בהכרח 0. בפתרון לבעיה 3 ערכו מאותחל ב-0 ובפתרון לבעיה 4 ערכו מאותחל ב-20. בגוף הלולאה משתנה ערכו של הצובר בערך כלשהו, השונה מ-1 (שימו לב כי ייתכן שערכו של הצובר עלול לקטון במהלך הצבירה, למשל, במקרה של צבירת ערכים שליליים).

נציג כעת את מאפייני שתי התבניות. ראשית, נציג את מאפייני התבנית **מנייה** ואחר כך נציג את מאפייני התבנית **צבירה**. עבור כל אחת מהתבניות נראה אלגוריתם עבור ביצוע חוזר באורך הידוע מראש וכן אלגוריתם לביצוע חוזר בתנאי.

שם התבנית: מנייה

נקודת מוצא: אורך סדרת נתוני הקלט limit, סדרת ערכי הקלט, תנאי מנייה condition

מטרה: מנייה של ערכי הקלט המקיימים את התנאי condition, מתוך סדרת קלט שאורכה limit

אלגוריתם:

1. אגף אף count 0-2

2. כצע limit פעמים:

2.1 קלוט ערך 2-element

2.2 אף element מקיים אף condition

2.2.1 הצף אף count 1-2

יישום ב-C#:

```
count = 0;
for (i = 1; i <= limit; i++)
{
    element = int.Parse(Console.ReadLine());
    if (condition)
    {
        count++;
    }
}
```

שם התבנית: מנייה

נקודת מוצא: תנאי סיום conditionToEnd, ערכי הקלט, תנאי מנייה conditionToCount  
מטרה: מנייה של ערכי הקלט המקיימים את התנאי conditiontoCount. משך ביצוע המנייה  
תלוי בתנאי conditionToEnd.

אלגוריתם:

1. אגף אף count 0-2

2. קאוט ערך 2-element

3. כן עזב לא מקיים conditionToEnd כ3ע:

3.1 אף element מקיים אף conditionToCount

3.1.1 הפז אף count 1-2

3.2 קאוט ערך 2-element

יישום ב-C#:

```
count = 0;
element = int.Parse(Console.ReadLine());
while (!conditionToEnd)
{
    if (conditionToCount)
    {
        count++;
    }
    element = int.Parse(Console.ReadLine());
}
```

ישנן שתי צורות בסיסיות של צבירה: צבירת סכום וצבירת מכפלה. אנו נראה תחילה את התבניות של צבירת סכום שמשך ביצועה ידוע מראש וצבירת סכום שמשך ביצועה תלוי בתנאי.

### שם התבנית: צבירת סכום

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערך תחילי של הצובר initial, ערכי הקלט, תנאי צבירה condition

**מטרה:** צבירת הסכום של הערך initial ושל סכום ערכי הקלט המקיימים את התנאי condition, מתוך סדרת קלט שאורכה limit

### אלגוריתם:

1. אגורל את sum ב-initial

2. בצע limit פעמים:

2.1 קולט ערך ב-element

2.2 אם element מקיים את condition

2.2.1 הוסף ל-sum את ערכו ב-element

### יישום ב-C#:

```
sum = initial;
for (i = 1; i <= limit; i++)
{
    element = int.Parse(Console.ReadLine());
    if (condition)
    {
        sum += element;
    }
}
```

## שם התבנית: צבירת סכום

**נקודת מוצא:** תנאי סיום conditionToEnd, ערך התחלתי של הצובר initial, ערכי הקלט, תנאי

צבירה conditionToSum

**מטרה:** צבירת סכום של הערך initial ושל ערכי הקלט המקיימים את התנאי conditionToSum.

משך ביצוע הצבירה תלוי בתנאי conditionToEnd.

### אלגוריתם:

1. אגור initial-2 sum

2. קאוט ערך-2 element

3. כן עוברים לא מקיים conditionToEnd כן ע3:

3.1 אם element מקיים את conditionToSum

3.1.1 הוסף ל-sum את ערכו element

3.2 קאוט ערך-2 element

### יישום ב-C#:

```
sum = initial;
element = int.Parse(Console.ReadLine());
while (!conditionToEnd)
{
    if (conditionToSum)
    {
        sum += element;
    }
    element = int.Parse(Console.ReadLine());
}
```

עתה נראה את התבנית של צבירת מכפלה שמשך ביצועה ידוע מראש ושל צבירת מכפלה שמשך ביצועה תלוי בתנאי.

**שם התבנית:** צבירת מכפלה

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערך התחלתי של הצובר initial, ערכי הקלט, תנאי צבירה condition

**מטרה:** צבירת מכפלה של הערך initial ושל ערכי הקלט המקיימים את התנאי condition, מתוך סדרת קלט שאורכה limit

**אלגוריתם:**

1. אגור את mult ב-initial

2. בצע limit פעמים:

2.1 קוטף ערך ב-element

2.2 אם element מקיים את condition

2.2.1 הכפול את mult ב-element והשם את המכפלה ב-mult

**יישום ב-C#:**

```
mult = initial;
for (i = 1; i <= limit; i++)
{
    element = int.Parse(Console.ReadLine());
    if (condition)
    {
        mult *= element;
    }
}
```

שם התבנית: צבירת מכפלה

נקודת מוצא: תנאי סיום conditionToEnd, ערך התחלתי של הצובר initial, ערכי הקלט, תנאי צבירה conditionToMult

מטרה: צבירת מכפלה של הערך initial ושל ערכי הקלט המקיימים את התנאי conditionToMult. משך ביצוע הצבירה תלוי בתנאי conditionToEnd.

אלגוריתם:

1. אגור initial-2 mult

2. קאוט ערך-2 element

3. כן עוברים לא מקיים conditionToEnd כן-ע:

3.1 אס element מקיים א conditionToMult

3.1.1 הכפל א mult-2 element והשם א המכפלה-2 mult

3.2 קאוט ערך-2 element

יישום ב-C#:

```
mult = initial;
element = int.Parse(Console.ReadLine());
while (!conditionToEnd)
{
    if (conditionToMult)
    {
        mult *= element;
    }
    element = int.Parse(Console.ReadLine());
}
```

**שימו** ♥: ערכו של initial חייב להיות שונה מ-0, שהרי מכפלת 0 בכל מספר תשאיר את ערכו של הצובר 0. ערכו של הצובר יכול להיות שלילי, וערכו של הצובר עשוי לקטון אם ערכו של element גדול מ-0 וקטן מ-1 (כלומר, הוא שבר), או אם נצברים גם ערכים שליליים.

**ועוד שימו** ♥: הבעיה האלגוריתמית אינה חייבת לכלול תנאים למנייה ולצבירה (condition), conditionToSum ו-conditionToMult. ייתכנו מקרים, כפי שאכן ראינו בבעיות האלגוריתמיות 2 ו-4, שהמנייה והצבירה מתבצעות ללא תנאי.

## שאלה 1

א. ישמו את פתרונותיהן של ארבע הבעיות האלגוריתמיות 1-4 בשפת C#.



- ב. בהתייחס לבעיה 1: שנו את התוכנית כך שתציג כפלט את מספר הילדים שאינם רשאים לעלות למתקן "רכבת הרים".
- ג. בהתייחס לבעיה 2: שנו את התוכנית כך שתציג כפלט את מספר הכניסות לאתר ביממה רק עבור המשתמשים שהקוד שלהם בין 100 ל-200.
- ד. בהתייחס לבעיה 3: הרחיבו את התוכנית כך שתציג כפלט גם את סכום המספרים השליליים.
- ה. בהתייחס לבעיה 4: הרחיבו את התוכנית כך שתציג כפלט גם את מספר המשקולות הכולל שהמתחרה הצליח להרים.

## שאלה 2

נתונה סדרת הקלט הבאה: 0 15 -6 14 -3 2

א. מה יוצג כפלט עבור כל אחד מהשימושים השונים בתבניות:

1. מנה את האיברים האי-חיוביים בסדרת הקלט שאורכה 6 והצג כפלט את הערך

שהקבל

2. מנה את האיברים הזוגיים בסדרת הקלט המסתיימת בזקוף 0 והצג כפלט את הערך

שהקבל

3. חשב את הסכום המצטבר של האיברים המתחלקים ב-3 בסדרת הקלט שאורכה 6

והצג כפלט את הערך שהקבל

4. חשב את הסכום המצטבר של 7 עם כל האיברים בסדרת הקלט המסתיימת בזקוף 0

והצג כפלט את הערך שהקבל

5. חשב את המכפלה המצטברת של האיברים האי-זוגיים בסדרת הקלט המסתיימת

בזקוף 5 והצג כפלט את הערך שהקבל

6. חשב את המכפלה המצטברת של 4 בכל האיברים בסדרת הקלט המסתיימת בזקוף

15 והצג כפלט את הערך שהקבל

ב. ישמו כל אחד מן השימושים בשפת C#.

## שאלה 3

א. בכיתה בת 41 תלמידים קיימו בחירות לנציג מועצת תלמידים. שני תלמידים הציגו את מועמדותם וכל תלמיד בכיתה הצביע עבור אחד המועמדים. נתון אלגוריתם שהקלט שלו הוא סדרה באורך 41 של המספרים 1 ו-2 המייצגים את מספרי המועמדים, והפלט שלו הוא מספרו של המועמד שזכה ברוב קולות.

1. אגף אגף count1-2-0

2. אגף אגף count2-2-0

3. כצט 41 פסמים:

3.1 קואוט מספרו של מועמד ב-candidate

3.2 אס candidate שווה ל-1

3.2.1 הגזא אס1 count1-2

3.3 אגרא

3.3.1 הגזא אס2 count2-1

4 אס count1 > count2

4.1 הגזא כפאט "מוסמז מספר 1 זכה ברוב קולות"

5 אגרא

5.1 הגזא כפאט "מוסמז מספר 2 זכה ברוב קולות"

א. באלגוריתם הנתון ישנו שילוב בשימוש של שני מונים: ציינו מהם ומהו תפקידם של אחד מהם.

ב. נתון אלגוריתם חלקי, השקול לאלגוריתם הנתון, אך משתמש במונה אחד. השלימו:

1. מנה את האיברים ה\_\_\_\_\_ בסדרת הקלט שאורכה 41 והגס count-2 אס

הערך המתקבל

2. אס \_\_\_\_\_

2.1 הגזא כפאט "מוסמז מספר 1 זכה ברוב קולות"

3. אגרא

3.1 הגזא כפאט "מוסמז מספר 2 זכה ברוב קולות"

#### שאלה 4

א. בסופרמרקט השכונתי בודקים מדי יום את הפדיון היומי ואת מספר הקונים. פתחו אלגוריתם שהקלט שלו הוא סדרת מספרים המייצגת את תשלומי הקונים, שבסיומה הזקיף 1-, והפלט שלו הוא מספר הקונים באותו יום והודעה המציינת אם ביום זה סך הפדיון מקניות בסכומים הגבוהים מ-500 ש"ח עלה על סך הפדיון מקניות בסכומים שאינם גבוהים מ-500 ש"ח. ישמו את האלגוריתם בשפת C#.

ב. ציינו באילו תבניות השתמשתם בסעיף א וכיצד **שילבתם** ביניהן.

#### שאלה 5

תת-סדרה תחילית של המספרים הטבעיים היא סדרה מהצורה  $1, 2, 3, \dots, n$ . בניסיון לבחון את "כוחה של המכפלה לעומת הסכום" הטילה המורה על תלמידיה לסכם את ערכיהן של תת-סדרות תחיליות של המספרים הטבעיים וכן להכפיל את ערכיהן.

א. פתחו אלגוריתם שאינו מקבל קלט, והפלט שלו הוא האורך הקטן ביותר של תת-סדרה תחילית של המספרים הטבעיים, שעבורה ערכה של המכפלה המצטברת יהיה לפחות פי 100 מערכו של הסכום המצטבר. ישמו את האלגוריתם בשפת C#.

- ב. בפתרון בעיה זו יש שימוש בצובר מכפלה. מהו ערכו התחילי של צובר זה? הסבירו.
- ג. ציינו באילו תבניות נוספות השתמשתם וכיצד **שילבתם** ביניהן.

## שאלה 6

לקראת בחינות בגרות בית הספר מעוניין לתת שיעורי תגבור ל-75 תלמידים משתי כיתות שונות. שיעורי התגבור יכולים להתקיים לאחר הלימודים, בימים ראשון ושלישי. כדי להיערך הוחלט להעביר שאלון, שבעזרתו ניתן יהיה לקבוע כמה תלמידים רוצים שיעורי תגבור ומהו היום המועדף על רובם. כל תלמיד רשם בשאלון את מספר כיתתו (1 או 2) ואת מספר היום שאותו הוא מעדיף (1 או 3). תלמיד שאינו מעוניין בתגבור רשם את מספר כיתתו ואת המספר 0.

כתבו אלגוריתם, שהקלט שלו הוא הערכים שרשמו 75 התלמידים, והפלט שלו כולל את:

1. מספר התלמידים המעוניינים בשיעורי תגבור.
  2. היום בשבוע שמרבית התלמידים מעדיפים.
  3. מספר החדרים הדרושים לשיעורי התגבור (בכל חדר ילמדו 15 תלמידים לכל היותר).
  4. מספר הכיתה שבה נרשמו פחות תלמידים לשיעורי התגבור.
- א. ישמו את האלגוריתם בשפת C#.

ב. ציינו באילו תבניות השתמשתם וכיצד **שילבתם** ביניהן.

## שאלה 7

א. פתחו אלגוריתם אשר מקבל כקלט סדרת גילאים של זוגות נשואים (גיל הבעל וגיל האישה). סוף הקלט יצוין על ידי זוג הערכים 0 0. הפלט יהיה אחוז הזוגות, שבהם גיל האישה גבוה מגילו של הבעל. ישמו את האלגוריתם בשפת C#.

ב. ציינו באילו תבניות השתמשתם וכיצד **שילבתם** ביניהן.

## שאלה 8

א. פתחו אלגוריתם אשר מקבל כקלט רשימת מספרים שלמים תלת-ספרתיים המסתיימת במספר שאינו תלת-ספרתי, והפלט שלו הוא כמות המספרים שספרת העשרות שלהם זוגית או שסכומן של ספרות המספר הוא אי-זוגי. ישמו את האלגוריתם בשפת C#.

ב. ציינו באילו תבניות השתמשתם וכיצד **שילבתם** ביניהן.

## ממוצע של סדרת מספרים

נתבונן בבעיה האלגוריתמית הבאה:

כתבו אלגוריתם שהקלט שלו כולל 31 מספרים המייצגים את הטמפרטורה בכל אחד מימי חודש יולי, והפלט שלו הוא ממוצע הטמפרטורות בחודש זה.

לפתרון הבעיה עלינו לחשב ממוצע של ערכים מספריים. בפרק 3 היכרנו את התבנית **ממוצע של סדרת מספרים**, עבור סדרה בת שני מספרים. עתה נרחיב את התבנית עבור סדרת מספרים באורך כלשהו. כדי לחשב ממוצע של סדרת מספרים יש לחשב תחילה את הסכום הכולל של איברי הסדרה ולאחר מכן לחלקו במספר הערכים בסדרה. התבנית של ממוצע, כפי שמשמשת בפתרון בעיה זו, דומה לחישוב הממוצע שהוצג בפתרון בעיה 3 בפרק הלימוד.

נתבונן בשני האלגוריתמים הבאים, הראשון לפתרון הבעיה שלעיל, והשני הוא האלגוריתם שניתן בפרק 7 כפתרון לבעיה 3:

1.  $\text{sum } a_n \text{ ב-} 0-2$

2. **כצד**  $\text{length}$  **כעמית**:

2.1. קאנט טמפרטורה של יום באודש יולי ב-temperature

2.2. הוסף ארכו של  $\text{sum } a_n$  ארכו של temperature

3. השם ב-average ארכו של הביטוי  $\text{sum}/31$  הגבול

4. הצג כפלט ארכו של average

1.  $\text{sum } a_n \text{ ב-} 0-2$

2.  $\text{counterLarge } a_n \text{ ב-} 0-2$

3. קאנט ערך גיובי שלם ב-length

4. **כצד**  $\text{length}$  **כעמית**:

4.1. קאנט ערך ממשי ב-num

4.2. הוסף ארכו של  $\text{sum}$  ארכו של num ארכו המצטבר השמו ב-sum

4.3. ארכו של  $\text{num}$  גבול מ-50

4.3.1. הצג ב-1 ארכו של  $\text{counterLarge}$

5. גשב ארכו ממוצע הערכים על ידי  $\text{sum}/\text{length}$  והשם ב-average

6. הצג כפלט ארכו של average

7. הצג כפלט ארכו של  $\text{counterLarge}$

בשני האלגוריתמים אותחל ערכו של sum ב-0 ואז חושב הסכום המצטבר בתוך sum. בתום הביצוע החוזר מתבצעת חלוקה של sum במספר הערכים שסוכמו. עבור הבעיה הנתונה מספר הערכים שסוכמו הוא 31, כמספר הימים בחודש יולי. בפתרון בעיה 3 שהוצגה בפרק הלימוד מספר הערכים המסוכמים הוא Length.

נציג את מאפייני התבנית לחישוב ממוצע עבור ביצוע חוזר התלוי בתנאי. שימו לב כי האלגוריתם של התבנית משלב במקביל תבנית צבירה (כדי לסכום את הערכים) ותבנית מנייה (כדי לדעת כמה ערכים נצברו).

#### שם התבנית: ממוצע של סדרת מספרים

**נקודת מוצא:** תנאי סיום conditionToEnd, ערכי הקלט, תנאי conditionToInclude

**מטרה:** חישוב ממוצע של ערכי הקלט המקיימים את התנאי conditionToInclude. ביצוע החישוב תלוי בתנאי conditionToEnd.

#### אלגוריתם:

1. אגף אף count 0-2
2. אגף אף sum 0-2
3. קלט ערך element-2
4. כן עדיף לא מקיים conditionToEnd **32**:
  - 4.1 אף element מקיים אף conditionToInclude
    - 4.1.1 הצד אף count 1-2
    - 4.1.2 הוסף לערכו אף sum אף element
  - 4.2 קלט ערך element-2
5. השם average אף ערכו אף הכיטוי הגשנוני sum / count

#### יישום ב-C#:

```
count = 0;
sum = 0;
element = int.Parse(Console.ReadLine());
while (!conditionToEnd)
{
    if (conditionToInclude)
    {
        count++;
        sum += element;
    }
    element = int.Parse(Console.ReadLine());
}
average = (double) sum / count;
```

בדומה לדרך בה נהגנו עבור תבניות **מנייה** ו-**צבירה** ניתן גם כאן להציג תת-תבנית המתאימה לביצוע חוזר שאורכו ידוע מראש. במקרה של ביצוע חוזר שאורכו ידוע מראש (והוא שווה ל-limit) אין צורך במניית הערכים. ולכן האלגוריתם הוא פשוט יותר:

1. חשב את הסכום המצטבר של איברי סדרת הקלט שאורכה limit

2. גזק את הסכום המצטבר של limit-2

**שימו** ♥: עבור התבנית של ממוצע של סדרת מספרים יש להניח שבקלט יש לפחות ערך אחד המקיים את התנאי, וזאת כדי להימנע מחלוקה ב-0.

**ועוד שימו** ♥: בדומה לתבניות **מנייה** ו-**צבירה**, גם בתבנית **ממוצע** ניתן להשתמש ללא תנאי (conditionToInclude), כלומר לחשב את ממוצע כל האיברים בסדרה.

---

## שאלה 9

א. יישמו את האלגוריתם לפתרון הבעיה של הטמפרטורה הממוצעת בחודש יולי בשפת C#.   
 ב. שנו את התוכנית שכתבתם בסעיף א כך שתחשב את הטמפרטורה הממוצעת רק עבור הימים שבהם הטמפרטורה הייתה מעל 30 מעלות צלסיוס.   
 ג. מעוניינים להשוות בין הטמפרטורה הממוצעת בחודש יולי בשנה זו לבין הטמפרטורה הממוצעת בחודש יולי בשנה הקודמת, ולהציג כפלט הודעה המציינת באיזו שנה מבין השתיים הטמפרטורה הממוצעת הייתה גבוהה יותר. הקלט כולל קודם כל את הערכים הנתונים עבור חודש יולי בשנה זו, ואחר כך את הנתונים עבור חודש יולי בשנה הקודמת. השלימו את האלגוריתם החלקי:

1. גזק את הממוצע של סדרת ערכי הקלט שאורכה \_\_\_\_\_ והשם **אלו** 2-  
averageCurrentYear

2. גזק את הממוצע של סדרת ערכי הקלט שאורכה \_\_\_\_\_ והשם **אלו** 2-  
averageLastYear

3. **אס** \_\_\_\_\_

3.1 **הצג כפלט** \_\_\_\_\_

4. **אגרג אס** \_\_\_\_\_

4.1 **הצג כפלט** \_\_\_\_\_

5. **אגרג**

5.1 **הצג כפלט** \_\_\_\_\_

## שאלה 10

באוניברסיטת "בית הסטודנט" הוחלט כי בקורס אשר בו ממוצע ציוני הבחינה של הסטודנטים הוא מתחת ל-55 ייערך מבחן חוזר. נתון אלגוריתם חלקי שהקלט שלו הוא סדרת ציוני הסטודנטים בקורס מסוים, רשימה המסתיימת בזקיף 1-, והפלט שלו הוא הודעה האם יש לקיים מבחן חוזר.

1. \_\_\_\_\_

2. \_\_\_\_\_

3. קאוט ערך 2-mark

4. כול ערך 3 mark שונה מ-1 - כ33:

4.1. \_\_\_\_\_

4.2. קאוט ערך 2-mark

5. השם 2-average את ערכו של הביטוי הגלובלי \_\_\_\_\_

6. אס \_\_\_\_\_

6.1. \_\_\_\_\_

א. השלימו את האלגוריתם.

ב. ניתן לכתוב את האלגוריתם בעזרת תבנית. השלימו:

1. \_\_\_\_\_ אס סדרת ערכי הקלט המסתיימת על ידי הזקיף 1-.

2. אס \_\_\_\_\_

2.1. \_\_\_\_\_

ג. ישמו את האלגוריתם בשפת C#.

## שאלה 11

בשירות המטאורולוגי מחשבים בכל שנה את ממוצע המשקעים החודשי.

א. כתבו אלגוריתם שהקלט שלו הוא 12 מספרים המייצגים את כמות המשקעים בכל אחד מחודשי השנה, והפלט שלו הוא ממוצע המשקעים לחודש.

ב. הרחיבו את האלגוריתם כך שיציג כפלט גם את ממוצע ששת החודשים הראשונים בשנה ואת ממוצע ששת החודשים האחרונים בשנה.

ג. ישמו את האלגוריתם בשפת C#.

## שאלה 12

במפעל מסוים התבקש מנהל מחלקת משאבי אנוש לבדוק האם גילן הממוצע של הנשים במפעל נמוך מגילם הממוצע של הגברים במפעל.

א. כתבו אלגוריתם שהקלט שלו הוא 100 זוגות נתונים המייצגים את מין העובד ('m' – גבר, 'f' – אישה) ושנת הלידה של העובד, והפלט שלו הוא הודעה מתאימה של מנהל מחלקת משאבי האנוש לאחר הבדיקה.

ב. הרחיבו את האלגוריתם שכתבתם בסעיף א כך שיציג כפלט את מספר העובדים הכולל שאמור לצאת לגמלאות ב-5 השנים הקרובות (הניחו כי גברים יוצאים לפנסיה בגיל 67 ונשים בגיל 62).

ג. ישמו את האלגוריתם בשפת C#.

ד. ציינו באילו תבניות השתמשתם וכיצד **שילבתם** ביניהן.

---



## מציאת מקסימום או מינימום בסדרה

נתבונן בבעיה האלגוריתמית הבאה:

למכרז זורמות הצעות כספיות שונות. ההצעה הכספית הגבוהה ביותר היא זו הזוכה במכרז. כתבו אלגוריתם שהקלט שלו הוא סדרה המסתיימת בזקיף 0, של מספרים המייצגים את ההצעות הכספיות, והפלט שלו הוא סכומה של ההצעה הזוכה. הניחו שבקלט יש לפחות הצעה אחת וכן הניחו שקיימת הצעה יחידה שזוכה.

לפתרון הבעיה עלינו לחשב את הערך המקסימלי מבין סדרת הערכים המספריים הנקלטים. בפרק 5 הראינו את התבנית של **מציאת מקסימום ומינימום בסדרה** בת שניים או שלושה ערכים. עתה נרחיב את התבנית עבור סדרה באורך כלשהו. התבנית של **מציאת מקסימום ומינימום בסדרה**, כפי שמשמשת בפתרון בעיה זו, דומה לתבנית **מציאת מקסימום ומינימום בסדרה** שהוצגה בבעיה 5 בפרק הלימוד.

נתבונן בשני האלגוריתמים הבאים, הראשון לפתרון הבעיה שלעיל, והשני הוא האלגוריתם שניתן בפרק 7 לפתרון בעיה 5:

1. קאוט הצעג מגיי כ-max

2. קאוט הצעג מגיי כ-price

3. כו ערז price  $\neq 0$  כצע:

3.1. אס price > max

3.1.1. השס כ-max אג ערכו של price

3.2. קאוט הצעג מגיי כ-price

4. הצג כפוט אג ערכו של max

---

1. קאוט אג מספרי הגשכונג כ-howMany

2. קאוט אג הסכום הראשון כ-balance

3. השס אג ערכו של max כ-balance

4. כצע howMany-1 כעמיס:

4.1. קאוט אג הסכום הכא כ-balance

4.2. אס max < balance

4.2.1. השס אג ערכו של max כ-balance

5. הצג כפוט אג ערכו של max

בשתי הבעיות נדרשנו למצוא את הערך הגדול ביותר בסדרת ערכים. בשלב ראשון, קבענו באופן שרירותי כי הערך של המשתנה הראשון הוא המקסימלי ולכן נשמר ערכו במשתנה max. לאחר מכן, נבדקו על פי סדר קליטתם ערכיהם של שאר נתוני הקלט. עבור כל נתון קלט שערכו גדול יותר מהמקסימום הנוכחי, הוחלף ערכו של max בערכו של נתון הקלט. באופן דומה, עם שינויים קלים, עובד האלגוריתם עבור מציאת הערך הקטן ביותר בסדרה.

נפריד את מאפייני התבנית **מציאת מקסימום ומינימום בסדרה** לשתי תת-תבניות: ראשית, נציג את מאפייני התבנית **מציאת מקסימום בסדרה** ואחר כך נציג את מאפייני התבנית **מציאת מינימום בסדרה**. עבור התבנית **מציאת מקסימום** נתייחס לביצוע חוזר באורך הידוע מראש ואילו עבור התבנית **מציאת מינימום** נתייחס לביצוע חוזר התלוי בתנאי. באופן דומה ניתן, עבור כל אחת מהתבניות, לטפל במקרה שאליו לא התייחסנו.

**שם התבנית:** מציאת מקסימום בסדרה

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערכי הקלט

**מטרה:** מציאת הערך הגדול ביותר בסדרת ערכי הקלט שאורכה הוא limit

**אלגוריתם:**

1. קאוט ערך  $max$ -2

2. כצט limit - 1 **פעמים:**

2.1. קאוט ערך  $element$ -2

2.2. אם  $element > max$

2.2.1. השם  $max$ -2 את הערך  $element$

**יישום ב-C#:**

```
max = int.Parse(Console.ReadLine());
for (i = 2; i <= limit; i++)
{
    element = int.Parse(Console.ReadLine());
    if (element > max)
    {
        max = element;
    }
}
```

**שם התבנית:** מציאת מינימום בסדרה

**נקודת מוצא:** תנאי סיום conditionToEndMinSearch, ערכי הקלט

**מטרה:** מציאת הערך הקטן ביותר מבין ערכי הקלט, עד אשר מתקיים התנאי

conditionToEndMinSearch

**אלגוריתם:**

1. קאוט ערך min-2

2. קאוט ערך element-2

3. כל עוד לא מתקיים conditionToEndMinSearch **כצע:**

3.1 אם  $element < min$

3.1.1 השם min-2 את הערך element

3.2 קאוט ערך element-2

**יישום ב-C#:**

```
min = int.Parse(Console.ReadLine());
element = int.Parse(Console.ReadLine());
while (!conditionToEndMinSearch)
{
    if (element < min)
    {
        min = element;
    }

    element = int.Parse(Console.ReadLine());
}
```

**שימו** ♥: לפני ההוראה לביצוע חוזר כבר נקראים שני ערכים מהקלט. אנו נניח שהתנאי לסיום (conditionToEndMinSearch) יכול להתקיים עבור האיבר השני ואילך. לכן, למשל, אם תנאי הסיום מבטא קריאת זקיף, הרי שסדרת הקלט מכילה לפחות איבר אחד.

**ועוד שימו** ♥: אם בקלט יש שני ערכים או יותר השווים למקסימום (או למינימום) אז האלגוריתמים עבור התבניות **מציאת מקסימום** ו-**מציאת מינימום** מוצאים את המופע הראשון של המקסימום (או המינימום) בנתוני הקלט.

### שאלה 13

א. השלימו את השימוש בתבנית עבור בעיית המכרז:

מצא מקסימום בסדרת ערכי הקלט \_\_\_\_\_ והצג את ערכו כפלט

ב. ישמו את האלגוריתם של המכרז בשפת C#.

ג. הרחיבו את האלגוריתם כך שיוצג כפלט גם סכום ההצעה השנייה בגודלה. ישמו את האלגוריתם המתקבל בשפת C#.

### שאלה 14

מדענים העוסקים בתחום המטאורולוגיה החליטו לגלות מהי הטמפרטורה הנמוכה ביותר ומהי הטמפרטורה הגבוהה ביותר בצהרי היום בחודש נובמבר באזור הקוטב הצפוני, בו הטמפרטורה תמיד מתחת ל-0 מעלות צלסיוס. לצורך כך, המדענים מדדו את הטמפרטורה במשך 30 יום במהלך חודש נובמבר, מדי יום ביומו, בשעה 12:00 בצהריים.

נתון אלגוריתם שגוי, שהקלט שלו הוא 30 ערכי הטמפרטורות והפלט שלו הוא הערך הנמוך ביותר והערך הגדול ביותר:

1. אגף `maxTemperature` - 0

2. אגף `minTemperature` - 0

3. כצע 30 פעמים:

3.1 קאוט ערך טמפרטורה יומי - `temperature`

3.2 אס `temperature > maxTemperature`

3.2.1 השם - `maxTemperature` אגף `temperature`

3.3 אגף אס `temperature < minTemperature`

3.3.1 השם - `minTemperature` אגף `temperature`

4. הצג כפלט "הטמפרטורה הנמוכה ביותר באזור נובמבר היא"

`minTemperature`

5. הצג כפלט "הטמפרטורה הגבוהה ביותר באזור נובמבר היא"

`maxTemperature`

א. הסבירו במלים מדוע האלגוריתם שגוי.

ב. תקנו את האלגוריתם.

### שאלה 15

קבוצת אנשים מוגדרת כ"הומוגנית" אם טווח הגילאים של חבריה אינו עולה על 5 שנים, אחרת הקבוצה מוגדרת כ"הטרואוגנית".

א. כתבו אלגוריתם שהקלט שלו הוא מספר האנשים בקבוצה,  $n$ , ולאחר מכן סדרה של  $n$  מספרים המייצגים את גילאי החברים בקבוצה. הפלט שלו הוא הודעה האם הקבוצה "הומוגנית" או "הטרונגנית".

ב. ציינו באילו תבניות השתמשתם בכתיבת האלגוריתם וכיצד **שילבתם** ביניהן.

ג. ישמו את האלגוריתם בשפת C#.

---

## מציאת ערך נלווה למקסימום או מינימום בסדרה

היכרנו את התבנית של מציאת מקסימום או מינימום בסדרה אולם לעיתים אנו מעוניינים לאו דווקא בערכים של המקסימום והמינימום אלא, למשל, במיקומם היחסי בקלט. דוגמה למקרה כזה הוא מציאת היום בחודש ינואר בו ירדה כמות משקעים מקסימלית.

נתבונן בבעיה האלגוריתמית הבאה:

נרחיב את בעיית המכרז שהוצגה בסעיף הדין בתבנית **מציאת מקסימום ומינימום בסדרה** כך שהקלט הוא סדרה של זוגות נתונים, כאשר כל זוג מכיל את הקוד של מגיש ההצעה (מספר שלם) ואת סכום ההצעה הכספית. הסדרה מסתיימת עם קליטת הזקיף 0 כאיבר שני בזוג. הפלט של האלגוריתם הוא הקוד של מגיש ההצעה הזוכה.

לפתרון הבעיה עלינו לחשב את ההצעה המקסימלית מבין סדרת ההצעות הכספיות הנתונות בקלט, אבל אנו נדרשים בנוסף לשמור גם את קוד מגיש ההצעה. התבנית של **מציאת ערך נלווה למקסימום ולמינימום בסדרה**, כפי שמשמשת בפתרון בעיה זו, דומה לתבנית **מציאת ערך נלווה למקסימום ולמינימום בסדרה** שהוצגה בבעיה 6 בפרק הלימוד.

נתבונן בשני האלגוריתמים הבאים, האחד לפתרון הבעיה שלעיל, והשני הוא האלגוריתם שניתן בפרק 7 לפתרון בעיה 6:

1. קאוט קוד מגיש הצעה 2-winCode

2. קאוט הצעג מחיר 2-max

3. קאוט קוד נוסף של מגיש הצעה 2-code

4. קאוט הצעג מחיר נוספת 2-price

5. כול עוצר price  $\neq$  0 כצעה:

5.1 אס price > max

5.1.1 השם 2-max אג ערכו של price

5.1.2 השם 2-winCode אג ערכו של code

5.2 קאוט קוד מגיש הצעה 2-code

5.3 קאוט הצעג מחיר 2-price

6. הצעג כפאוט אג ערכו של winCode

1. קאוט אט אורק השיי הראשון ב- currentSongLength
2. השם אט ערכו של longest- currentSongLength
3. השם אט ערכו של shortest- currentSongLength
4. השם א- placeLongest אט הערך 1
5. השם א- placeShortest אט הערך 1
6. **ב3ע 99 פעמים :**

6.1 קאוט אט המספר הכא ב- currentSongLength

6.2 אט  $currentSongLength > longest$

6.2.1 השם אט longest- currentSongLength

6.2.2 השם אט מקומו של השיי הנכחי ב- placeLongest

6.3 אט  $currentSongLength < shortest$

6.3.1 השם אט shortest- currentSongLength

6.3.2 השם אט מקומו של השיי הנכחי ב- placeShortest

7. ה3ג כפאוט אט ערכו של placeLongest ואט ערכו של placeShortest

בשתי הבעיות נדרשנו למצוא ערך נלווה לערך הגדול ביותר בסדרת ערכים. בבעיה הנתונה בתחילת הסעיף הערך הנלווה הוא קוד מגיש ההצעה ובעיה 6 בפרק הלימוד הערך הנלווה הוא מקומו של השיר הארוך ביותר.

כדי למצוא את הערך הנלווה לערך הגדול ביותר בסדרת ערכים נתבסס על התבנית למציאת מקסימום: בכל פעם שנעדכן את ערכו של max (וכמובן גם באתחול) נשמור במשתנה נוסף את ערכו של הערך הנלווה ל-max. באופן דומה, עם שינויים קלים, מתבצע האלגוריתם עבור מציאת הערך הנלווה לערך הקטן ביותר בסדרה.

בתבנית של מציאת ערך נלווה למקסימום ולמינימום בסדרה נדגים כערך הנלווה את מקום המקסימום או המינימום, אבל כאמור, ערך נלווה יכול להיות כל ערך שהוא, כמו הקוד הנלווה להצעה, בבעיה שלעיל.

נפריד את מאפייני התבנית **מציאת ערך נלווה למקסימום ולמינימום בסדרה** לשתי תת-תבניות: ראשית, נציג את מאפייני התבנית **מציאת ערך נלווה למקסימום בסדרה** ואחר כך נציג את מאפייני התבנית **מציאת ערך נלווה למינימום בסדרה**. עבור התת-תבנית הראשונה נתייחס לביצוע חוזר באורך הידוע מראש, ואילו עבור השנייה נתייחס לביצוע חוזר התלוי בתנאי. משום הדמיון בין שתי התת-תבניות, קל לפתח אלגוריתם מתאים למקרה האחר, עבור כל אחת מהן.

**שם התבנית:** מציאת ערך נלווה למקסימום בסדרה

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערכי הקלט

**מטרה:** מציאת מיקום הערך הגדול ביותר בסדרת ערכי הקלט שאורכה הוא limit

**אלגוריתם:**

1. קלוט ערך max-2

2. אגף את placeOfMax ב-1

3. בצע limit-1 פעמים:

3.1. קלוט ערך element-2

3.2. אם  $element > max$

3.2.1. השם ב-max את הערך של element

3.2.2. השם ב-placeOfMax את מקומו של element בקלט

**יישום ב-C#:**

```
max = int.Parse(Console.ReadLine());  
placeOfMax = 1;  
for (i = 2; i <= limit; i++)  
{  
    element = int.Parse(Console.ReadLine());  
    if (element > max)  
    {  
        max = element;  
        placeOfMax = i;  
    }  
}
```



**שם התבנית:** מציאת ערך נלווה למינימום בסדרה

**נקודת מוצא:** תנאי סיום conditionToEnd, ערכי הקלט

**מטרה:** מציאת מיקום הערך הקטן ביותר מבין ערכי הקלט, עד אשר מתקיים התנאי

conditionToEnd

**אלגוריתם:**

1. קאט ערך min-2

2. אגף אף placeOfMin-2

3. קאט ערך element-2

4. אגף אף currentPlace-2

5. כן עזב לא מתקיים conditionToEnd כ-3:

5.1 אף element < min

5.1.1 השט min-2 אף הערך של element

5.1.2 השט min-2 placeOfMin אף הערך של currentPlace

5.2 הגדף אף currentPlace-2

5.3 קאט ערך element-2

**יישום ב-C#:**

```
min = int.Parse(Console.ReadLine());
placeOfMin = 1;
element = int.Parse(Console.ReadLine());
currentPlace = 2;
while (!conditionToEnd)
{
    if (element < min)
    {
        min = element;
        placeOfMin = currentPlace;
    }
    currentPlace++;
    element = int.Parse(Console.ReadLine());
}
```

**שימו** ♥ : לפני ההוראה לביצוע חוזר כבר נקראים שני ערכים מהקלט. אנו נניח שהתנאי לסיום (conditionToEnd) יכול להתקיים עבור האיבר השני ואילך. לכן, למשל, אם תנאי הסיום מבטא קריאת זקיף, הרי שסדרת הקלט מכילה לפחות איבר אחד.

לא פעם מופיע הערך המינימלי או המקסימלי בסדרה יותר מפעם אחת. בתבנית המוצגת מחושב מקום המופע **הראשון** של הערך המקסימלי או המינימלי. שינוי מזערי באלגוריתם יאפשר את חישוב מקום המופע **האחרון**. שאלה 16 מתייחסת לנקודה זאת. עם ההתקדמות בחומר הלימוד נתייחס גם לסוגיית מציאת **כל** המופעים של הערך המקסימלי או המינימלי.

---

### שאלה 16

א. ישמו את האלגוריתם המורחב של המכרז בשפת C#.

ב. עתה, הניחו שבקלט יש שתי הצעות זוכות, כלומר שהערך הגבוה ביותר של הצעה הוצע על ידי שני מגישים. שנו את האלגוריתם, ואחר כך את התוכנית שכתבתם בסעיף א, כך שיוצג כפלט הקוד של מגיש ההצעה השנייה (במקום של הראשונה).

### שאלה 17

חנוכייה אופיינית היא חנוכייה בת 9 קנים, שאחד מהם גבוה מן השאר ומשמש כשמש. חנוכייה כזו נחשבת "סימטרית" אם השמש ממוקם במרכז החנוכייה (ארבעה קנים ממוקמים מימינו של השמש וארבעה ממוקמים משמאלו). חנוכייה כזו נחשבת "צידיית" אם השמש ממוקם בקצה אחד שלה (שאר הקנים ממוקמים משמאלו של השמש, או מימינו). בכל מקרה אחר, נחשבת החנוכייה "מיוחדת".

א. כתבו אלגוריתם שהקלט שלו הוא 9 מספרים המייצגים את גבהי הקנים של חנוכייה אופיינית, ונתונים לפי סדר מיקומם (משמאל לימין), והפלט שלו הוא הודעה המציינת את סוג החנוכייה.

ב. ישמו את האלגוריתם בשפת C#.

### שאלה 18

בשכבת כיתות י' נערך מבחן משווה במדעי המחשב. המבחן נבדק על ידי המורים, וציוני התלמידים נרשמו בטופס ריכוז בו כל תלמיד מיוצג על ידי מספר סידורי. כל מורה התבקש למסור למרכז המגמה את הציון הגבוה ביותר בכיתה ומספרו הסידורי של התלמיד שקיבל ציון זה, וכן את הציון הנמוך ביותר בכיתה ומספרו הסידורי של התלמיד שקיבל ציון זה (המורה ידווח על תלמיד אחד מכל קטגוריה, גם אם יש יותר מתלמיד אחד שקיבל את הציון הגבוה ביותר בכיתה או יותר מתלמיד אחד שקיבל את הציון הנמוך ביותר בכיתה).

א. כתבו אלגוריתם שהקלט שלו הוא מספר התלמידים n בכיתה מסוימת, ואחריו סדרה של n ציוני התלמידים בכיתה, והפלט שלו הוא הציון הגבוה ביותר בכיתה ומספרו הסידורי של התלמיד שקיבל ציון זה, והציון הנמוך ביותר בכיתה ומספרו הסידורי של התלמיד שקיבל ציון זה.

ב. הרחיבו את האלגוריתם כך שיציג כפלט גם את הציון השני בגודלו.

ג. ישמו את האלגוריתם בשפת C#.

## איסוף בקיזוז

נתבונן בבעיה האלגוריתמית הבאה:

באוניברסיטה נערך סקר לבדיקת ההשערה: "מספר הסטודנטיות הלומדות קורסים המשלבים מחשב קטן יותר ממספר הסטודנטים בקורסים אלה". כתבו אלגוריתם שהקלט שלו הוא סדרה באורך 100 שאיבריה הם התו 'f' או התו 'm', המייצגת את מינם ('m'-סטודנט, 'f'-סטודנטית) של 100 סטודנטים הלומדים קורסים המשלבים מחשב. הפלט הוא הודעה האם ההשערה נכונה או אינה נכונה ביחס לנתונים.

בפתרון הבעיה אין חשיבות למספר הסטודנטים ומספר הסטודנטיות הלומדים קורסים המשלבים מחשב. לכן אין צורך במנייה רגילה. במקרה זה מתאים יותר למנות תוך כדי קיזוז. כלומר, נגדיר מונה יחיד, שערכו יגדל בכל פעם שנקלוט סטודנט, וערכו יקטן בכל פעם שנקלוט סטודנטית. בסיום התהליך יהיה במונה ההפרש בין מספר הסטודנטים למספר הסטודנטיות. נבדוק את ערכו של המונה: אם ערכו חיובי ההשערה נכונה. אחרת, ההשערה אינה נכונה. גם עבור שאלה 7.27 בפרק הלימוד ניתן לכתוב אלגוריתם המתבסס על איסוף בקיזוז.

נתבונן בשני האלגוריתמים הבאים, האחד לפתרון הבעיה שלעיל, והשני לפתרון שאלה 7.27:

1. אגוא אג 0-2 count	1. אגוא אג 0-2 count
2. קאוס קוא בואי 2-vote	2. כצצ 100 קעמאיס:
3. כא עוצ #' < vote כצצ:	2.1. קאוס מין הסטואיני 2-gender
3.1. אקס אקס וואה ל-'A'	2.2. אקס אקס וואה ל-'m'
3.1.1. אג אג אג 1-2 count	2.2.1. אג אג אג 1-2 count
3.2. אגרא	2.3. אגרא
3.2.1. אג אג אג 1-2 count	2.3.1. אג אג אג 1-2 count
3.3. קאוס קוא בואי 2-vote	3. אקס אקס אג 0-M count
4. אקס אקס אג 0-M count	3.1. אג אג אג "ההשערה נכונה"
4.1. אג אג אג "אגון זכה"	4. אגרא
4.2. אג אג אג "אגון לא זכה"	4.1. אג אג אג "ההשערה לא נכונה"

התבנית של איסוף בקיזוז מתאימה גם לקיזוז באמצעות צבירה ולא רק לקיזוז באמצעות מנייה. בהמשך נראה שאלות, שבהן יש צורך לאסוף בקיזוז באמצעות צבירה.

נפריד את מאפייני התבנית **איסוף בקיזוז** לשתי תת-תבניות: ראשית, נציג את מאפייני התבנית **איסוף בקיזוז באמצעות מנייה** ואחר כך נציג את מאפייני התבנית **איסוף בקיזוז באמצעות צבירה**. עבור התבנית הראשונה נתייחס לביצוע חוזר באורך הידוע מראש ואילו עבור השנייה נתייחס לביצוע חוזר התלוי בתנאי. כבמקרים קודמים, עבור כל אחת מהתבניות ניתן לפתח אלגוריתמים מתאימים גם למקרה האחר.

**שם התבנית: איסוף בקיזוז באמצעות מנייה**

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערכי הקלט, תנאי לקיזוז conditionToDecrease

**מטרה:** איסוף בקיזוז באמצעות מנייה של ערכי הקלט מתוך סדרה שאורכה limit. הקיזוז מתבצע על פי התנאי conditionToDecrease

**אלגוריתם:**

1. אגף אף count 0-2

2. בצע limit פעמים:

2.1 קאוט ערך element-2

2.2 אם element לא מקיים אף conditionToDecrease

2.2.1 הגדל אף count 1-2

2.3 אגף

2.3.1 הקטן אף count 1-2

**יישום ב-C#:**

```
count = 0;
for (i = 1; i <= limit; i++)
{
    element = int.Parse(Console.ReadLine());
    if (!conditionToDecrease)
    {
        count++;
    }
    else
    {
        count--;
    }
}
```

**שימו** ♥ : בתבנית **איסוף בקיזוז** התנאי condition הוא הכרחי, ותפקידו, שליטה בקיזוז, שונה מתפקיד התנאים conditionToCount, conditionToSum, או conditionToMult בתבניות **מנייה ו-צבירה**.

עתה נראה את התבנית של איסוף בקיזוז באמצעות צבירה, עבור שמשך ביצועה תלוי בתנאי.

**שם התבנית: איסוף בקיזוז באמצעות צבירה**

**נקודת מוצא:** תנאי סיום conditionToEnd, ערכי הקלט, ערך צבירה התחלתי initial, תנאי לקיזוז conditionToDecrease

**מטרה:** איסוף בקיזוז באמצעות צבירה של ערכי הקלט ושל הערך initial, משך הביצוע תלוי בתנאי conditionToEnd, והקיזוז מתבצע על פי התנאי conditionToDecrease

**אלגוריתם:**

1. אגור initial-2 sum

2. קאוט ערך-2 element

3. כן עזב לא מקיים conditionToEnd כ3ע:

3.1. אס element לא מקיים א conditionToDecrease

3.1.1. הוסף-1 sum א ערכו א element

3.2. אגר

3.2.1. הפג-1 sum א ערכו א element

3.3. קאוט ערך-2 element

**יישום ב-C#:**

```
sum = initial;
element = int.Parse(Console.ReadLine());
while (!conditionToEnd)
{
    if (!conditionToDecrease)
    {
        sum += element;
    }
    else
    {
        sum -= element;
    }
    element = int.Parse(Console.ReadLine());
}
```

## שאלה 19

ישמו את הפתרון לבעיית הסקר באוניברסיטה בשפת C#.

## שאלה 20

נתונה סדרת ההוראות הבאה, שיש בה שימוש בתבנית איסוף בקיזוז :

1. אסוף בקיזוז באמצעות מנייה את איברי סדרת הקלט שאורכה 8, תוך קיזוז

האיברים האי-זוגיים

2. הצג כפלט את הערך המתקבל

א. תנו דוגמה לסדרת נתוני קלט של מספרים חיוביים שעבורה יוצג כפלט הערך 3.

ב. הסבירו בקצרה מהי מטרת ההוראה.

ג. השלימו את סדרת ההוראות הבאה, השקולה לסדרה הנתונה, אך משתמשת בתנאי לקיזוז

אחר :

1. אסוף בקיזוז באמצעות מנייה את איברי סדרת הקלט שאורכה 8, תוך קיזוז

האיברים הזוגיים

2. הצג כפלט את \_\_\_\_\_

## שאלה 21

נתון אלגוריתם שהקלט שלו הוא סדרת מספרים ממשיים :

1. אגף את sum ב-0

2. קאוט מספר ממש ב-num

3. כן אם  $num \neq 0$  אחרת לא:

3.1 אם  $num > 0$

3.1.1 הוסף ל-sum את ערכו של num

3.2 אגף

3.2.1 הפגף מ-sum את ערכו של num

3.3 קאוט מספר ב-num

4. הצג כפלט את ערכו של sum

א. תנו שתי דוגמאות קלט שונות לסדרות נתוני קלט של מספרים ממשיים, שעבורן יוצג כפלט

הערך 0.

ב. הסבירו בקצרה מהי מטרת האלגוריתם.

ג. כתבו הוראה השקולה לאלגוריתם תוך שימוש בתבנית. השלימו :

אסוף בקיזוז באמצעות צבירה ל- \_\_\_\_\_ של איברי סדרת הקלט המסתיימת

בצקיף 0, תוך קיזוז \_\_\_\_\_

## שאלה 22

- א. כתבו אלגוריתם שהקלט שלו הוא יתרת לקוח של בנק בתחילת החודש, ואחריו סדרה של מספרים, המסתיימת ב-0. כל מספר מייצג את סכום הפעולה: סכום חיובי מציין הפקדת הסכום בחשבון הבנק וסכום שלילי מציין משיכת הסכום מחשבון הבנק. הפלט של האלגוריתם הוא היתרה של הלקוח בסוף החודש.
- ב. ישמו את האלגוריתם בשפת C#.

## שאלה 23 (שאלה זו מתאימה לאחר לימוד סעיף 7.7 – קינון הוראות לביצוע חוזר)

- מבחן במתכונת של שאלון אמריקני הועבר ל-30 סטודנטים של הקורס "דיוק בתשובות". במבחן 20 שאלות. ניקוד תשובות התלמידים התבצע באופן הבא: כל תשובה נכונה זיכתה ב-5 נקודות וכל תשובה שגויה גרמה להפחתה של 3 נקודות (בכל מקרה, לא ניתן לקבל ציון נמוך מ-0).
- א. כתבו אלגוריתם, שהקלט שלו הוא 20 התשובות הנכונות ואחריהן תשובות של 30 הסטודנטים בקורס (20 תשובות לכל סטודנט). הפלט הוא מספר הסטודנטים שציונם "עובר" (לפחות 55).
- ב. הרחיבו את האלגוריתם שכתבתם בסעיף א כך שיציג כפלט גם את אחוז התלמידים שקיבלו 0 במבחן.
- ג. הרחיבו את האלגוריתם שכתבתם בסעיף ב כך שיציג כפלט גם את הציון הגבוה ביותר בבחינה ואת הציון הנמוך ביותר בבחינה (שאינו 0).
- ד. ציינו מהן התבניות המשמשות לפתרון וכיצד שילבתם ביניהן.
- ה. ישמו את האלגוריתם שכתבתם בסעיף ג בשפת C#.

## שאלה 24 (\*)

נתונים ארבעה ביטויים בוליאניים המשתמשים בתבניות, הבודקים אותו תנאי, עבור סדרת קלט נתונה:

1. *השך המגבל מ-איסוף בקיזוז באמצעות צבירה של איברי סדרת הקלט שאורכה 10, תוך קיזוז האיברים הזוגיים האז זוגי*
2. *השך המגבל מ-מנייה של האיברים האי-זוגיים בסדרת הקלט שאורכה 10 האז זוגי*
3. *השך המגבל מ-צבירת סכום איברי סדרת הקלט שאורכה 10 האז זוגי*
4. *השך המגבל מ-צבירת סכום האיברים האי-זוגיים בסדרת הקלט שאורכה 10 האז זוגי*

- א. תנו דוגמה לסדרת מספרים שלמים חיוביים שעבורה הערך של כל אחד מהביטויים הבוליאניים יהיה true.
- ב. נסחו במלים את התנאי שמבטאים הביטויים הבוליאניים.
- ג. ישמו כל אחד מהשימושים השונים כקטעי תוכניות בשפת C#.

## פירוק מספר חיובי לספרותיו

בפרק 4 הכרנו את התבנית של פירוק מספר דו-ספרתי חיובי לספרותיו. עתה נרחיב את התבנית עבור מספר שלם חיובי באורך כלשהו. פירוק המספר מתבסס על רעיון של הפרדת ספרת האחדות מהמספר, כך שהמספר שנוותר קטן פי 10, וחוזר חלילה עד שלא נותרות ספרות במספר.

נראה את מאפייני התבנית **פירוק מספר חיובי לספרותיו**:

<p><b>שם התבנית:</b> פירוק מספר חיובי לספרותיו</p> <p><b>נקודת מוצא:</b> מספר שלם חיובי num</p> <p><b>מטרה:</b> הצגה כפלט של ספרותיו של num</p> <p><b>אלגוריתם:</b></p> <ol style="list-style-type: none"><li>1. כל עוד num שונה מ-0 כצע</li><li>2. הצג כפלט את ספרת האחדות של num</li><li>3. הקטן את num פי 10</li></ol> <p><b>יישום ב-C#:</b></p> <pre>while (num != 0) {     Console.WriteLine(num % 10);     num /= 10; }</pre>
---

**שימו ♥:** באלגוריתם של התבנית הצגנו כפלט את ספרותיו של num, אולם ניתן כמובן לבצע פעולות אחרות על ספרות המספר, כגון: מנייה, צבירה, ועוד. השאלות הבאות מתייחסות לבעיות אלגוריתמיות בהן נדרשות פעולות אחרות על ספרות המספר.

### שאלה 25

- פתחו אלגוריתם שהקלט שלו הוא מספר שלם חיובי, והפלט שלו הוא מספר הספרות במספר. ישמו את האלגוריתם בשפת C#.
- הרחיבו את האלגוריתם שכתבתם בסעיף א כך שיציג כפלט גם את סכום הספרות האי-זוגיות במספר.
- הרחיבו את האלגוריתם שכתבתם בסעיף ב כך שיציג כפלט גם הודעה המציינת אם יש במספר יותר ספרות המתחלקות ב-3 ללא שארית מאשר ספרות שאינן מתחלקות ב-3. ישמו את האלגוריתם המלא בשפת C#.
- ציינו באילו תבניות השתמשתם בכתיבת האלגוריתם המלא וכיצד **שילבתם** ביניהן.



## שאלה 26

לפניכם קטע תוכנית בשפת C#, שהקלט שלו הוא מספר שלם חיובי, והפלט שלו אמור להיות מספר הספרות במספר. קטע התוכנית שגוי.

```
num = int.Parse(Console.ReadLine());
sum = num % 10;
while ((num / 10) > 0)
    sum += num % 10;
Console.WriteLine(sum);
```

- הביאו דוגמת קלט שעבורה יתקבל הפלט הדרוש.
- הביאו דוגמת קלט שעבורה לא יתקבל הפלט הדרוש.
- הסבירו במלים מדוע קטע התוכנית שגוי.
- תקנו את קטע התוכנית כך שישגי את מטרתו עבור כל קלט חוקי.

## שאלה 27

א. כתבו אלגוריתם שהקלט שלו הוא מספר שלם וחיובי num ומספר חד-ספרתי place המייצג מקום. הפלט שלו הוא הספרה הנמצאת במקום place במספר num (מיקום הספרות מתחיל מימין). אם אין במספר place ספרות יוצג כפלט הערך -1.

למשל, עבור הקלט 2 17489 יוצג כפלט הערך 8, ועבור הקלט 6 17489 יוצג כפלט הערך -1.

- ישמו את האלגוריתם בשפת C#.

## שאלה 28

ספרת שורש של מספר היא ספרה בין 1 ל-9 המתקבלת מתהליך של חיבור חוזר של ספרות המספר עד אשר מתקבל מספר חד-ספרתי. למשל, ספרת השורש של המספר 30486 היא 2, כיוון שסכום ספרות המספר המקורי הוא 20 וסכום הספרות של 20 הוא 2.

א. כתבו אלגוריתם שהקלט שלו הוא מספר שלם חיובי והפלט שלו הוא ספרת השורש של המספר הנתון.

- ישמו את האלגוריתם בשפת C#.

## שאלה 29

א. כתבו אלגוריתם שהקלט שלו הוא מספר שלם חיובי והפלט שלו הוא ההפרש בין הספרה הגדולה ביותר במספר לבין הספרה הקטנה ביותר במספר.

ב. ציינו באילו תבניות השתמשתם בכתיבת האלגוריתם וכיצד **שילבתם** ביניהן.

- ישמו את האלגוריתם בשפת C#.

## בניית מספר

בפרק 4 הכרנו את התבנית של בניית מספר כאשר התמקדנו בבניית מספר דו-ספרתי משתי ספרות. עתה נרחיב את התבנית לבניית מספר שלם באורך כלשהו מספרות הנקלטות בזו אחר זו. כדי להרכיב מספר מספרות צריך בכל שלב בלולאה להגדיל את המספר פי 10 ולחבר לו את ספרת הקלט החדשה. כך עד לסיום הקלט.

ניתן להרחיב את התבנית, למשל, על ידי בניית מספר ממספרים דו-ספרתיים, ואז בכל שלב יגדל המספר פי 100 ויתווסף לו המספר הדו-ספרתי התורן מהקלט.

נציג עבור התבנית **בניית מספר** אלגוריתם עבור ביצוע חוזר באורך הידוע מראש. ניתן להתאימו גם למקרים בהם סיום הבנייה תלוי בתנאי.

**שם התבנית: בניית מספר**

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ספרות הקלט

**מטרה:** בניית מספר מספרות הקלט

**אלגוריתם:**

1. אגף אג num 0-2
2. כצע limit פעמים:
  - 2.1 קאט ספרה 2-digit
  - 2.2 השם 2-num אג הערך של הביטוי הגשולי  $num * 10 + digit$

**יישום ב-C#:**

```
num = 0;
for (i = 1; i <= limit; i++)
{
    digit = int.Parse(Console.ReadLine());
    num = num * 10 + digit;
}
```

### שאלה 30

נתון האלגוריתם הבא:

1. קאט ספרה 2-digit
2. קאט מספר 2-limit
3. אגף אג num 0-2
4. כצע limit פעמים:

4.1. השם  $num$ -2 הערך של הביטוי  $num * 10 + digit$

4.2. הערך כפול  $num$

- מה יהיה הפלט עבור הקלט 3 5?
- תנו דוגמה לקלט שעבורו הפלט יהיה 6.
- הסבירו בקצרה מהי מטרת האלגוריתם.

### שאלה 31

- א. כתבו אלגוריתם שהקלט שלו הוא מספר שלם וחיובי  $num$ , ספרה  $digit$  ומקום  $place$ . הפלט שלו הוא המספר שמתקבל מהחלפת הספרה הנמצאת במקום  $place$  (מימין) במספר  $num$  בספרה  $digit$ .
- למשל, עבור הקלט 6 4 78342 יוצג כפלט הערך 76342, ועבור הקלט 9 1 13608 יוצג כפלט הערך 13608.
- ב. ישמו את האלגוריתם בשפת C#.

### שאלה 32

- א. כתבו אלגוריתם שהקלט שלו הוא מספר שלם וחיובי  $num$  והפלט שלו הוא המספר המתקבל מ- $num$  על ידי היפוך סדר ספרותיו.
- ב. ציינו באילו תבניות השתמשתם בכתיבת האלגוריתם וכיצד **שילבתם** ביניהן.
- ג. ישמו את האלגוריתם בשפת C#.
-

## האם כל הערכים בסדרה מקיימים תנאי?

נתבונן בבעיה האלגוריתמית הבאה:

במסגרת הפעילות של עידוד "הקורא הצעיר" הוחלט להעניק פרס לבית-ספר שבו כל תלמידיו קראו לפחות ספר קריאה אחד במהלך חופשת הקיץ. פתחו אלגוריתם שהקלט שלו הוא מספר הספרים שקרא כל אחד מתלמידי בית-ספר "אמירים" שבסיומה הזקיף 1- (עבור תלמיד שלא קרא כלל ספרים ייקלט הערך 0), והפלט שלו הוא הודעה המציינת אם הפרס יוענק לבית-הספר.

לפתרון הבעיה עלינו לעבור על כל נתוני הקלט, כלומר, לבדוק עבור כל תלמיד האם קרא ספרי קריאה במהלך החופשה או לא. אבל, אם קיים לפחות תלמיד אחד שלא קרא ספר קריאה אז לבית-הספר לא יוענק הפרס ולכן אין טעם להמשיך ולבדוק את מספר ספרי הקריאה שקראו שאר תלמידי בית-הספר. אלגוריתם זה הוא תיאור של התבנית **האם כל הערכים בסדרה מקיימים תנאי?** . גם עבור שאלה 7.53 בפרק הלימוד ניתן לכתוב אלגוריתם המתבסס על תבנית זו.

נתבונן בשני האלגוריתמים הבאים, הראשון לפתרון הבעיה שלעיל, והשני לפתרון שאלה 7.53:

1. אגף א `allReaders` ב-`true`
2. קאנט מספר ספרי קריאה אגף ב-`books`
3. כן ע"כ `books ≠ 1` / `אם ערכו של allReaders הוא true` כן ע"כ:
  - 3.1. אם ערכו של `books` הוא 0
    - 3.1.1. השם ב-`allReaders` אגף הערך `false`
    - 3.2. אגף א
      - 3.2.1. קאנט מספר ספרי קריאה אגף ב-`books`
  4. אם ערכו של `allReaders` הוא `true`
    - 4.1. הציג כפאנט "לביא-הספר אמירים מושנק הפרס"

- 
1. אגף א `allEven` ב-`true`
  2. אגף א `sum` ב-0
  3. אגף א `howMany` ב-1
  4. כן ע"כ `howMany ≤ 20` / `אם ערכו של allEven הוא true` כן ע"כ:
    - 4.1. קאנט מספר גיובי ב-`num`
    - 4.2. הציג אגף ערכו של `howMany` ב-1
    - 4.3. אם `num` מספר אי-זוגי
      - 4.3.1. השם ב-`allEven` אגף הערך `false`
      - 4.4. אגף א
        - 4.4.1. הציג אגף `sum` ב-`num`

5. אס allEven

5.1. הציג כפאזט אג ערכו של sum

6. אגרא

6.1. הציג כפאזט "הקאזט אינו אוקי"

משמעות התבנית **האם כל הערכים בסדרה מקיימים תנאי?** היא בדיקה של קיום תנאי עבור כל ערכי הסדרה. מתבצעת סריקה של ערכי הקלט בזה אחר זה, ואם אחד הערכים אינו מקיים את התנאי אין טעם בהמשך הסריקה. בבעיה הנתונה אם ערכו של books שווה ל-0 אז נמצא תלמיד שלא קרא אף לא ספר קריאה אחד ולכן אין צורך להמשיך בסריקה. בשאלה 7.53 אם בקלט יש מספר אי-זוגי אז אין טעם להמשיך בבדיקת שאר ערכי המספרים.

נציג את מאפייני התבנית **האם כל הערכים בסדרה מקיימים תנאי?** עבור סדרה שאורכה ידוע מראש. ניתן להתאים את האלגוריתם למקרה שבו אורך הסדרה אינו ידוע מראש, בדומה לאלגוריתם שניתן לבעיה שלעיל.

**שם התבנית:** האם כל הערכים בסדרה מקיימים תנאי?

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערכי הקלט, תנאי condition

**מטרה:** קביעת הערך true אם כל הערכים בסדרה מקיימים את התנאי condition וקביעת הערך false אם קיים ערך אחד בסדרה שאינו מקיים את התנאי

**אלגוריתם:**

1. אגרא אג all ב-true

2. אגרא אג howmany ב-1

3. כן אם  $howmany \leq limit$  / אג ערכו של all הוא true ב-33:

3.1. קאזט ערך ב-element

3.2. הציג אג ערכו של howMany ב-1

3.3. אס element אינו מקיים אג condition

3.3.1. השט ב-all אג הערך false

**יישום ב-C#:**

```
all = true;
howmany = 1;
while ((howmany <= limit) && (all))
{
    element = int.Parse(Console.ReadLine());
    howmany++;
    if (condition) // התנאי
    {
        all = false;
    }
}
```

התבנית **האם כל הערכים בסדרה מקיימים תנאי?** מחשבת ערך בוליאני שערכו true או false. אנו משתמשים במשתנה בוליאני all שערכו התחילי הוא true, כלומר, ההנחה **התחילית** היא שכל ערכי הסדרה אכן מקיימים את התנאי. אם במהלך הסריקה אחד הערכים אינו מקיים את התנאי אז ההנחה התחילית שלנו מתבדה ולכן ערכו של all מקבל ערך false ו הסריקה מסתיימת.

---

### שאלה 33

א. רשמו הוראה השקולה לאלגוריתם של "הקורא הצעיר" תוך שימוש בתבנית החדשה. השלימו:  
**האם כל הערכים בסדרת הקלט המסתיימת ב-\_\_\_\_\_ מקיימים את התנאי'\_\_\_\_\_?**

ב. ישמו את האלגוריתם בשפת C#.

### שאלה 34

א. כתבו אלגוריתם שהקלט שלו הוא מספר שלם וחיובי num והפלט שלו הוא ההודעה "ספרות זהות" אם כל ספרות המספר זהות וההודעה "ספרות שונות" אם לא כל הספרות זהות.  
ב. ציינו באילו תבניות השתמשתם בכתיבת האלגוריתם וכיצד **שילבתם** ביניהן.  
ג. ישמו את האלגוריתם בשפת C#.

### שאלה 35

לפניכם קטע תוכנית בשפת C#, שהקלט שלו הוא סדרה של מספרים שלמים וחיוביים. סוף הקלט מצוין על ידי המספר -1. קטע התוכנית אמור להציג כפלט את ההודעה "כל המספרים הם כפולות של 6" אם 6 מחלק של כל המספרים בסדרת הקלט. קטע התוכנית שגוי.

```
num = int.Parse(Console.ReadLine());  
while (num != -1)  
{  
    ok = (num % 6 == 0)  
    num = int.Parse(Console.ReadLine());  
}  
if (ok)  
{  
    Console.WriteLine("All numbers are multiples of 6");  
}
```

- א. תנו דוגמה לסדרת קלט (לפחות 5 ערכים) עבורה לא ניתן להבחין שקטע התוכנית שגוי.  
ב. תנו דוגמה לסדרת קלט (לפחות 5 ערכים) עבורה ניתן להבחין שקטע התוכנית שגוי.  
ג. הסבירו במילים מדוע קטע התוכנית שגוי.  
ד. תקנו את קטע התוכנית כך שיבצע את מטרתו עבור כל סדרת קלט.

## האם קיים ערך בסדרה המקיים תנאי?

נתבונן בשתי הבעיות האלגוריתמיות הבאות:

**בעיה 1:** כתבו אלגוריתם, שהקלט שלו הוא סדרה של 15 מספרים שלמים, והפלט שלו הוא הודעה המציינת אם קיים בסדרת הקלט מספר שלילי.

**בעיה 2:** כתבו אלגוריתם, שהקלט שלו הוא סדרה של תווים, המסתיימת בזקיף '\*', והפלט שלו הוא הודעה המציינת אם נקלטה בסדרת התווים אחת מאותיות ה-ABC או אחת מאותיות ה-abc.

לפתרון שתי הבעיות עלינו לעבור על נתוני הקלט עד למציאת ערך המקיים את התנאי. בבעיה 1 התנאי הוא האם ערך הקלט הוא מספר שלילי ובבעיה 2 התנאי הוא האם נקלטה אחת מאותיות ה-ABC או אחת מאותיות ה-abc. אם קיים ערך אחד המקיים את התנאי אז יש להפסיק את הסריקה כי ערך מתאים כבר נמצא. אלגוריתם זה הוא תיאור של התבנית **האם קיים ערך בסדרה המקיים תנאי?**

נתבונן בשני האלגוריתמים לפתרון בעיה 1 ובעיה 2:

1. אגף אגף found-2 false

2. אגף אגף howmany-2 1

3. כן ע"כ  $15 \leq \text{howmany}$  / אגף ערכו של found שווה ל-1 false כ"ע:

3.1 קאוט מספר שלם כ-2 num

3.2 הג"א אגף ערכו של howmany-2 1

3.3 אס  $\text{num} < 0$

3.3.1 השם כ-2 found אגף הערך true

4. אס found

4.1 הצג כפאט "קיים מספר שלילי בסדרת הקלט"

5. אגף

5.1 הצג כפאט "לא קיים מספר שלילי בסדרת הקלט"

1. אגף אגף found-2 false

2. קאוט ג' כ-2 ch

3. כן ע"כ  $\text{ch} \neq '*'$  / אגף ערכו של found הוא false כ"ע:

3.1 אס  $(ch \leq 'z' \wedge ch \geq 'a')$  / א  $(ch \leq 'Z' \wedge ch \geq 'A')$

3.1.1 השם found-2 אג הערך true

3.2 אגרא

3.2.1 קאוט גא-2 ch

4 אס found

4.1 הצג כפוט "קיימת אור-2 ABC או abc בסדרה הקוט"

5 אגרא

5.1 הצג כפוט "לא קיימת אור-2 ABC או abc בסדרה הקוט"

נציג את מאפייני התבנית האם קיים ערך בסדרה המקיים תנאי? עבור סדרה שאורכה אינו ידוע מראש. ניתן להתאים את האלגוריתם למקרה שבו אורך הסדרה ידוע מראש, בדומה לאלגוריתם שניתן לבעיה 1 שלעיל.

**שם התבנית:** האם קיים ערך בסדרה המקיים תנאי?

**נקודת מוצא:** תנאי לסיום הסדרה toEnd, ערכי הקלט, תנאי condition

**מטרה:** קביעת הערך true אם קיים ערך בסדרה המקיים את התנאי condition וקביעת הערך

false אם כל הערכים בסדרה אינם מקיימים את התנאי

**אלגוריתם:**

1. אגרא אג found-2 false

2. קאוט ערך-2 element

3. כן עוצר הגנאי toEnd לא מקיים אגם ערכו של found הוא false כצג:

3.1 אס element מקיים אג condition

3.1.1 השם found-2 אג הערך true

3.2 אגרא

3.2.1 קאוט ערך-2 element

**יישום ב-C#:**

```
found = false;
element = int.Parse(Console.ReadLine());
while (!toEnd && !found)
{
    if (condition) // התנאי
    {
        found = true;
    }
    element = int.Parse(Console.ReadLine());
}
```



משמעות התבנית **האם קיים ערך בסדרה המקיים תנאי?** היא בדיקה של קיום תנאי עבור ערכים בסדרה. מתבצעת סריקה של ערכי הקלט בזה אחר זה, ואם אחד הערכים אכן מקיים את התנאי אין טעם בהמשך הסריקה. התבנית מחשבת ערך בוליאני שערכו true או false. לכן, אנו משתמשים במשתנה בוליאני found שערכו התחילי הוא false. כלומר, ההנחה **התחילית** היא שאין ערך בסדרה המקיים את התנאי. אם במהלך הסריקה אחד הערכים אכן מקיים את התנאי אז ההנחה התחילית שלנו מתבדה, מאחר שמצאנו ערך המקיים את התנאי. לכן found יקבל ערך true והסריקה תסתיים.

---

### שאלה 36

א. כתבו הוראה השקולה לאלגוריתמים של כל אחת מהבעיות 1 ו-2 תוך שימוש בתבנית **האם קיים ערך בסדרה המקיים תנאי?**. השלימו:

בעיה 1:

1. **אם קיים ערך בסדרת הקלט שאורכה \_\_\_\_\_ המקיים \_\_\_\_\_**

1.1 **הצג כפאט "קיים מספר שלילי בסדרת הקלט"**

2. **אגרא**

2.1 **הצג כפאט "לא קיים מספר שלילי בסדרת הקלט"**

בעיה 2:

1. **אם קיים ערך בסדרת הקלט שמסתיימת ב- \_\_\_\_\_ המקיים \_\_\_\_\_**

1.1 **הצג כפאט "קיימת אלף ב-ABC אף ב-abc בסדרת הקלט"**

2. **אגרא**

2.1 **הצג כפאט "לא קיימת אלף ב-ABC אף ב-abc בסדרת הקלט"**

ב. ישמו את האלגוריתמים לפתרון הבעיות 1 ו-2 בשפת C#.

ג. עבור כל אחד מהאלגוריתמים ניתן לכתוב אלגוריתם שקול תוך שימוש בתבנית **האם כל**

**הערכים בסדרה מקיימים תנאי?**. השלימו את ההוראות הבאות עבור בעיה 1, וכתבו

הוראות מתאימות, המשתמשות בתבנית זו, לבעיה 2.

בעיה 1:

1. **אם לא כל הערכים בסדרה שאורכה \_\_\_\_\_ מקיימים את התנאי**

1.1 **הצג כפאט "קיים מספר שלילי בסדרת הקלט"**

2. **אגרא**

2.1 **הצג כפאט "לא קיים מספר שלילי בסדרת הקלט"**

### שאלה 37

א. כתבו אלגוריתם, שהקלט שלו מספר שלם חיובי והפלט שלו הוא הודעה המציינת אם קיימת במספר הספרה 0.

ב. שנו את האלגוריתם שכתבתם בסעיף א כך שיציג הודעה המציינת אם קיימת ספרה זוגית במספר. ציינו איזו תבנית **שילבתם** בפתרון הבעיה.

### שאלה 38

בכיתה י' המורה העלתה הצעה לעשות קומוזיץ לגיבוש 39 תלמידי הכיתה. ההצעה תתקבל רק אם אף תלמיד לא יתנגד להגיע לקומוזיץ. כל תלמיד התבקש לרשום על דף את האות 'y' אם הוא מסכים להצעה או את האות 'n' אם הוא מתנגד להצעה. יש לכתוב אלגוריתם, שהקלט שלו הוא סדרה של 39 תווים כאשר התו 'y' מייצג את הסכמת התלמיד להצעה והתו 'n' מייצג את התנגדות התלמיד להצעה. הפלט הוא הודעה המציינת אם ההצעה התקבלה או לא.

לפניכם שני אלגוריתמים לפתרון הבעיה:

אלגוריתם 1:

1. **אם קיים ערך בסדרת הקלט שאורכה 39 המקיים את התנאי (answer שווה ל-**

**'n')**

1.1 **הצג כפלט את ההודעה "ההצעה לא התקבלה"**

2. **אגור**

2.1 **הצג כפלט את ההודעה "ההצעה התקבלה"**

אלגוריתם 2:

1. **אם כל הערכים בסדרת הקלט שאורכה 39 מקיימים את התנאי (answer שווה**

**ל-'y')**

1.1 **הצג כפלט את ההודעה "ההצעה התקבלה"**

2. **אגור**

2.1 **הצג כפלט את ההודעה "ההצעה לא התקבלה"**

ישמו את שני האלגוריתמים בשפת C#.

## מציאת כל הערכים בסדרה המקיימים תנאי

נתבונן בשתי הבעיות האלגוריתמיות הבאות:

**בעיה 1:** כתבו אלגוריתם, שהקלט שלו הוא סדרה של 25 זוגות מספרים שלמים, והפלט שלו הוא כל זוגות המספרים המקיימים את היחס של מספרים עוקבים.

**בעיה 2:** כתבו אלגוריתם, שהפלט שלו הוא כל המספרים מ-1 עד 100 המקיימים את הכללים של "7 בוס", כלומר: מתחלקים ב-7 ללא שארית או כוללים את הספרה 7.

לפתרון שתי הבעיות עלינו לעבור על כל הערכים בסדרה ולהציג כפלט את כל הערכים המקיימים את התנאי. בבעיה 1 התנאי הוא יחס של עוקב בין כל זוג מספרים ברשימה ובבעיה 2 התנאי הוא קיום הכללים של "7 בוס". יש דמיון מסוים בין תבנית זו לשתי התבניות האחרונות, אך במקרה זה איננו יכולים להפסיק את הסריקה לפני שנגיע אל סיום סדרת הקלט. אלגוריתם זה הוא תיאור של התבנית **מציאת כל הערכים בסדרה המקיימים תנאי**.

נתבונן בשני האלגוריתמים לפתרון בעיה 1 ובעיה 2:

1. כצג 25 פסאים:

2. קאוט צג מספרים שאינם  $num1$  ו- $num2$

2.1. אס  $num1$  ו- $num2$  הם ערכים עוקבים א  $num1$  ו- $num2$  הם ערכים

עוקבים

2.1.1. הצג כפוט אג הערכים א  $num1$  ושל  $num2$

---

1. עבור כל מספר  $i$  גיובי אס הקטן מ-100 כצג:

1.1. אס 7 מחלק של  $i$  א ספרת העשרות של  $i$  שווה ל-7 א ספרת האחדות

של  $i$  שווה ל-7

1.1.1. הצג כפוט אג הערך א  $i$

בתבנית **מציאת כל הערכים בסדרה המקיימים תנאי** מתבצעת סריקה של כל ערכי הקלט בזה אחר זה. בכל פעם שנמצא ערך שמקיים את התנאי מבצעים עליו פעולה כגון הצגה כפלט, מנייה, פעולה חשבונית.

נציג את מאפייני התבנית **מציאת כל הערכים בסדרה המקיימים תנאי** עבור סדרה שאורכה אינו ידוע מראש. ניתן להתאים את האלגוריתם למקרה שבו אורך הסדרה ידוע מראש, בדומה

לאלגוריתם שניתן לבעיה 1 ו-2 שלעיל. מאפייני התבנית מוגדרים לפי ביצוע פעולת קלט עבור כל איבר שנמצא מקיים את התנאי. ניתן להתאימם למקרה בו נדרשת פעולה אחרת, כפי שקורה בכמה מהשאלות הבאות.

**שם התבנית:** מציאת כל הערכים בסדרה המקיימים תנאי

**נקודת מוצא:** תנאי לסיום הסדרה toEnd, ערכי הקלט, תנאי condition

**מטרה:** הצגה כפלט של כל ערכי הקלט המקיימים את התנאי condition

**אלגוריתם:**

1. קאוט ערך 2-element

2. כן עזי הגאוי toEnd לא מקיים 23ע:

3.1 אס element מקיים אג condition

3.1.1 הצג כפלט אג ערכו על element

3.2 קאוט ערך 2-element

**יישום ב-C#:**

```
element = int.Parse(Console.ReadLine());  
while (!toEnd)  
{  
    if (condition)  
    {  
        Console.WriteLine(element);  
    }  
    element = int.Parse(Console.ReadLine());  
}
```

### שאלה 39

א. רשמו הוראה השקולה לאלגוריתם לפתרון בעיה 1 תוך שימוש בתבנית. השלימו:  
מצא את כל הערכים בסדרת הקלט שאורכה \_\_\_\_\_ המקיימים את התנאי

ב. ישמו את האלגוריתם בשפת C#.

ג. רשמו הוראה השקולה לאלגוריתם לפתרון בעיה 2 תוך שימוש בתבנית. השלימו:  
מצא את כל הערכים בסדרת הקלט שאורכה \_\_\_\_\_ המקיימים את התנאי

ד. ישמו את האלגוריתם בשפת C#.

#### שאלה 40

תלמידי שכבה י' בבית-ספר "היובל" השתתפו בתחרויות יום ספורט היתולי, במהלכו כל משתתף צבר נקודות. תוצאתו של תלמיד שלא השתתף היא 0. למען גיבוש הכיתות הוחלט להעניק "יום כיף" לכיתות שבהן השתתפו כל התלמידים בתחרויות של יום הספורט. נתון אלגוריתם חלקי, שהקלט שלו הוא תוצאות התחרויות של כל אחת מ-10 הכיתות בבית-הספר. לכל כיתה נקלטת סדרת התוצאות של כל התלמידים עד לקליטת הזקיף 1-. הפלט של האלגוריתם הוא מספרי הכיתות שזכו ב"יום כיף".

1.  $i$  גיוכי ולס הקטן א שווה ל-10  $z3$ :

1.1. אגא א prize-2- \_\_\_\_\_

1.2. קאט גוצא א גאמיז 2-result \_\_\_\_\_

1.3. כא  $z3$  \_\_\_\_\_ אג \_\_\_\_\_  $z3$ :

1.3.1. אג \_\_\_\_\_

1.3.1.1. false prize-2- אג העניק

1.3.2. אגא \_\_\_\_\_

1.3.2.1. \_\_\_\_\_

2. אג \_\_\_\_\_

2.1. העז כפאט \_\_\_\_\_

א. השלימו את האלגוריתם.

ב. ציינו מהן התבניות המשולבות באלגוריתם.

ג. בהנהלת בית-הספר החליטו להעניק פרס נוסף "גיבושון" לכיתה שתלמידיה צברו את מירב הנקודות בתחרות. הרחיבו את האלגוריתם כך שיציג כפלט את מספר הכיתה שזכתה בפרס "גיבושון". איזו תבנית שילבתם עתה בפתרון הבעיה האלגוריתמית?

ד. ישמו את האלגוריתם בשפת C#.

#### שאלה 41

א. כתבו אלגוריתם, שהקלט שלו הוא סדרה של מספרים חיוביים (סוף הסדרה מצוין ע"י מספר שלילי), והפלט שלו הוא כל המספרים המתחלקים בסכום ספרותיהם וכמות המספרים המקיימים תנאי זה.

לדוגמה: עבור סדרת הקלט 7- 18 103 83 110 550 4000: המספרים 18 550 110 ו-1

4000 מתחלקים בסכום ספרותיהם, לכן הפלט הוא רשימת המספרים 18 550 110 4000

והודעה המציינת כי כמות המספרים המתחלקים בסכום ספרותיהם: 4

ב. ישמו את האלגוריתם בשפת C#.

## מעבר על זוגות סמוכים בסדרה

נתבונן בשתי הבעיות האלגוריתמיות הבאות:

**בעיה 1:** כתבו אלגוריתם, שהקלט שלו הוא סדרה של 18 מספרים שלמים, והפלט שלו הוא הודעה האם הסדרה מסודרת בסדר עולה ממש.

**בעיה 2:** כתבו אלגוריתם, שהקלט שלו הוא סדרה של מספרים שלמים חיוביים, המסתיימת עם קליטת הזקיף 0, והפלט שלו הוא סדרה שבה מכפלות כל זוגות המספרים הזוגיים הנקלטים זה אחר זה בסמיכות. לדוגמה, עבור הקלט: 0 2 4 8 9 4 5 3 תוצג כפלט הסדרה: 8 32 24, המתקבלת ממכפלת 6 ב-4, מכפלת 4 ב-8 ומכפלת 4 ב-2.

לפתרון שתי הבעיות יש לעבור על כל זוגות הערכים הסמוכים בסדרה. בבעיה 1 יש לבדוק עבור כל זוג ערכים סמוכים אם הוא מקיים את היחס של הערך הראשון (משמאל) קטן יותר מהערך השני. בבעיה 2 יש להציג כפלט את המכפלות של זוגות מספרים סמוכים שאיבריהם זוגיים. זה הוא, למעשה, תיאור של התבנית **מעבר על זוגות סמוכים בסדרה**.

נתבונן בשני האלגוריתמים הבאים לפתרון בעיות 1 ו-2:

1. אגף אג `true-2 ordered`
2. אגף אג `1-howmany`
3. קאוס מספרי אג `beforeLast-2`
4. כן `howmany < 18` / `אג סוכו` `ordered` / `אג`
  - 4.1. קאוס מספרי אג `last-2`
  - 4.2. אג `beforelast ≥ last`
    - 4.2.1. השם `ordered-2` אג `false`
    - 4.3. אגף אג
    - 4.3.1. השם `beforeLast-2` אג `last`
5. אג `ordered`
  - 5.1. הצג כפאוס אג `ההוצעה "הסדרה מסודרת בסדר עולה אמ"`
6. אגף אג
  - 6.1. הצג כפאוס `"הסדרה אינה מסודרת בסדר עולה אמ"`

- 
1. קאוס מספרי אג `beforeLast-2`
  2. קאוס מספרי אג `last-2`
  3. כן `last < 0` `כצג`:

3.1. **אם** beforeLast הוא מספר זוגי **אם** last הוא מספר זוגי

3.1.1. **הצג** כפולט את הערך של הביטוי הגולמי beforeLast \* last

3.2. **השם** beforeLast-2 את הערך של last

3.3. **קולט** מספר שלם beforeLast-2

בפתרון בעיות אלגוריתמיות רבות יש לבצע פעולות על זוגות ערכים הסמוכים בסדרה. עבור סדרת נתוני קלט באורך limit יש בסך הכול limit-1 זוגות ערכים סמוכים. למעשה ניתן להסתכל על כל זוג ערכים סמוכים כאל פריט אחד. הרעיון שבבסיסם של שני האלגוריתמים הוא שבכל שלב המשתנים beforeLast ו-last מכילים איברי זוג סמוך. המעבר לזוג הבא מתבצע כך: beforeLast מקבל את ערכו של last (כלומר, האיבר שקודם היה איבר שני בזוג הוא עכשיו איבר ראשון בזוג הסמוך) וב-last נקלט ערך נוסף. כך עד לסיום הקלט.

לשם המחשה נדגים בתבנית הצגה כפלט של סכום זוגות ערכים סמוכים בסדרה. ניתן כמובן לבצע פעולות אחרות על ערכי כל זוג, כגון מנייה, צבירה ועוד. בהמשך נדגים כמה מהאפשרויות השונות דרך שאלות.

נציג את מאפייני התבנית **מעבר על זוגות סמוכים בסדרה** עבור סדרה שאורכה ידוע מראש. אפשר לערוך התאמות למקרה בו אורך סדרת הקלט אינו ידוע מראש, כמו בבעיה 2 לעיל.

**שם התבנית:** מעבר על זוגות סמוכים בסדרה

**נקודת מוצא:** אורך סדרת נתוני הקלט limit, ערכי הקלט

**מטרה:** הצגה כפלט של סכומי כל זוגות הערכים הסמוכים בסדרת הקלט שאורכה limit

**אלגוריתם:**

1. **קולט** ערך beforeLast-2

2. **בצע** limit-1 פעמים

2.1. **קולט** ערך beforeLast-2

2.2. **הצג** כפולט את הערך של הביטוי הגולמי beforeLast + last

2.3. **השם** beforeLast-2 את הערך של last

**יישום ב-C#:**

```
beforeLast = int.Parse(Console.ReadLine());  
for (i = 1; i <= limit-1; i++)  
{  
    last = int.Parse(Console.ReadLine());  
    Console.WriteLine(beforeLast + last);  
    beforeLast = last;  
}
```

**שימו** ♥ : אנו מניחים שבסדרת הקלט לפחות שני ערכים.

---

#### שאלה 42

ישמו את האלגוריתמים של שתי הבעיות הנתונות בשפת C# .

#### שאלה 43

א. כתבו אלגוריתם, שהקלט שלו הוא 25 תווים והפלט שלו הוא כל הזוגות הסמוכים בסדרת הקלט שמכילים אותיות קטנות ב-abc עוקבות. ישמו את האלגוריתם בשפת C# .  
ב. ציינו באילו תבניות נוספות השתמשתם בכתיבת האלגוריתם וכיצד **שילבתם** ביניהן.

#### שאלה 44

כתבו אלגוריתם שהקלט שלו הוא סדרת מספרים שלמים המסתיימת בזקיף 1-, והפלט הוא כל המספרים בקלט השווים לסכום שני המספרים הקודמים להם בסדרה (שני המספרים הראשונים לא יודפסו).  
למשל, עבור סדרת הקלט הבאה (משמאל לימין): 1- 89 6 5 13 0 13 11 2 9 4 יוצגו כפלט המספרים: 13 13 11.  
ישמו את האלגוריתם בשפת C# .

#### שאלה 45

כתבו תוכנית בשפת C#, שהקלט שלה הוא מספר שלם וחיובי num והפלט שלה הוא הודעה האם ספרות המספר מסודרות בסדר עולה ממש.  
ציינו באילו תבניות נוספות השתמשתם בכתיבת התוכנית וכיצד **שילבתם** ביניהן.

---