

תבניות – פרק 10

מערך מונים

נתבונן בבעיה האלגוריתמית הבאה:

כתבו אלגוריתם שהקלט שלו הוא 58 ספרות והפלט שלו הוא מספר הפעמים שהופיעה כל ספרה בקלט.

למעשה, עבור פתרון הבעיה האלגוריתמית אנו צריכים 10 מונים – מונה לכל ספרה. כדי להימנע מסירבול, נגדיר מערך `CountDigitsArr` בגודל 10. מצייני המערך הם מ-0 ועד 9, ולכן כל תא במערך מייצג את המונה של המציין. כך, למשל, תפקידו של התא `countDigitsArr[3]` הוא למנות את מספר הפעמים שמופיעה הספרה 3 בקלט. באופן כללי תפקידו של התא `countDigitsArr[digit]` הוא למנות את מספר הפעמים שמופיעה הספרה `digit` בקלט. זהו אלגוריתם המתבסס על התבנית **מעריך מונים**. התבנית של **מעריך מונים**, כפי שמשמשת בחלקו הראשון של פתרון בעיה זו, דומה לחלקו הראשון של הפתרון שהוצג בבעיה 5 בפרק 10. נתבונן בחלק הראשון של כל אחד משני האלגוריתמים הללו, לפתרון הבעיה שלעיל ולפתרון בעיה 5 בפרק 10:

1. כצד 58 פעמים:

1.1 קאנט ספריה 2-digit

1.2 הגדל אג `countDigitsArr[digit]` 1-2

1. קאנט אג מספרי האמאודיים 2-numOfSingers

2. קאנט מספרי מושאד 2-vote

3. כול עוזד 1-vote \neq כצד:

3.1 העלה 2-1 אג האולה של אמאוד מספרי vote

3.2 קאנט מספרי מושאד 2-vote

בשני קטעי האלגוריתמים האלה ניתן לזהות תבנית **מנייה**, אלא שהיא מופעלת על כמה מונים הפועלים במקביל, ושמורים במערך אחד.

נציג את מאפייני התבנית **מעריך מונים**, עבור ביצוע חוזר התלוי בתנאי. ניתן להתאים את מאפייני התבנית למקרים בהם משך הביצוע ידוע מראש, בדומה לקטע האלגוריתם לפתרון הבעיה שהוצגה לעיל.

שם התבנית: מערך מונים

נקודת מוצא: תנאי סיום toEnd, סדרת ערכים, ביטוי חשבוני whichCounter המקשר בין ערך בסדרה למציין של המערך

מטרה: בניית מערך מונים עבור ערכי הסדרה, בעוד משך הבנייה תלוי בביטוי toEnd.

אלגוריתם:

1. אגף את איברי המערך countElements 0-2

2. השם את הערך הכא כסדרה ב-element

3. כן עזר לא מקיים הנאי toEnd כצט:

3.1 הגדף את countElements[whichCounter] 1-2

3.2 השם את הערך הכא כסדרה ב-element

יישום ב-C#:

```
הערך הבא בסדרה = element;  
while (!toEnd)  
{  
    countElements[whichCounter]++;  
    הערך הבא בסדרה = element;  
}
```

שימו ♥:

- ♦ התבנית כללית ואינה מפרטת מהיכן מגיעים איברי הסדרה. ערכים אלה יכולים, למשל, להתקבל כקלט או מקריאת ערכי מערך אחר.
- ♦ המשתנה element מתייחס לאיבר התורן בסדרה. למרות שאין זה מופיע במפורש באלגוריתם שניתן עבור התבנית וביישום שלו, הרי פעולת המנייה תלויה ב-element: הביטוי whichCounter תלוי ב-element ומקשר בין ערכו לערך של מציין במערך. הנה כמה דוגמאות לביטויים אפשריים עבור whichCounter:
 - element (כלומר, הערך הבא בסדרה משמש כמציין למערך, בדומה לפתרון של הבעיה שהוצגה בתחילת הסעיף).
 - ספרת העשרות של element.
 - 1 - element (בדומה לביטוי המשמש לגישה למערך בפתרון בעיה 5 בפרק 10).
- ♦ בדומה לתבנית מנייה, גם התבנית של מערך מונים כוללת אתחול של המונים ל-0. אין לכך התייחסות ביישום, משום שבשפת C# מתבצע אתחול אוטומטי של איברי מערך שלמים ל-0.

שאלה 1

- א. ישמו את הפתרון המלא לבעיית מניית מופעי הספרות בשפת C#.
- ב. שנו את התוכנית שכתבתם בסעיף א כך שהקלט שלה יהיה 58 מספרים מהתחום 1 עד 10, והפלט שלה יהיה מספר הפעמים שהופיע כל אחד מהמספרים. מהו הביטוי החשבוני בו השתמשתם כדי לקשר בין ערך תורן למציין במערך?
- ג. שנו את התוכנית שכתבתם בסעיף א כך שהקלט שלה יהיה 58 מספרים שלמים חיוביים. הפלט שלה יהיה עבור כל אחת מהספרות 0 עד 9 את מספר הפעמים שהופיעה כספרת האחדות של מספר בקלט. מהו הביטוי החשבוני בו השתמשתם כדי לקשר בין ערך תורן למציין במערך?

שאלה 2

נתון המערך `arr` ובו ערכים שלמים ונתון קטע התוכנית הבא:

```
int[] counts = new int[2];
for (i = 0; i < arr.length; i++)
{
    element = arr[i];
    counts[element % 2]++;
}
for (i = 0; i < counts.length; i++)
{
    Console.WriteLine(counts[i]);
}
```

א. מה יוצג כפלט עבור המערך `arr` הבא:

<code>arr[0]</code>	<code>arr[1]</code>	<code>arr[2]</code>	<code>arr[3]</code>	<code>arr[4]</code>	<code>arr[5]</code>	<code>arr[6]</code>	<code>arr[7]</code>
90	68	198	5	11	34	89	6

- ב. תנו דוגמה למערך `arr` בגודל 10, שעבורו יוצגו כפלט הערכים: 3 7.
- ג. מהי מטרת קטע התוכנית?
- ד. הציעו קטע תוכנית השקול לקטע התוכנית הנתון **ללא** שימוש בתבנית **מערך מונים**. ציינו באילו תבניות השתמשתם עבור כתיבת קטע התוכנית.

שאלה 3

- א. כתבו אלגוריתם, שהקלט שלו הוא סדרת תווים המסתיימת בתו '*' והפלט שלו הוא מספר המופעים של כל אחת מאותיות ה-`abc` (כלומר, אותיות הא"ב האנגלי הקטנות).
- ב. מהו הביטוי החשבוני בו השתמשתם עבור התאמת אותיות ה-`abc` למצייני המערך?
- ג. ישמו את האלגוריתם בשפת C#.

מעריך צוברים

נתבונן בשתי הבעיות האלגוריתמיות הבאות:

בעיה 1: תלמידי שכבה י' בבית הספר "נוף-ים", המונה 7 כיתות, החליטו על מבצע לסיוע למשפחות נזקקות: כל תלמיד בשכבה יעבוד בעבודות מזדמנות (כגון שטיפת-מכוניות, בייביסיטר וכו'), ואת הסכום שירוויח יעביר לקופת הכיתה. בסוף החודש יעבירו הנציגים מכל אחת מהכיתות דיווח של הסכום המצטבר שנאסף אל נציג השכבה. נציג השכבה יעביר את הסכום הכולל לנציג ועד השכונה. יש לציין כי כל התלמידים בשכבה נרתמו להצלחת המבצע. כתבו אלגוריתם שהקלט שלו הוא 247 זוגות מספרים עבור כל תלמידי השכבה, כאשר המספר הראשון מייצג את מספר הכיתה של התלמיד והמספר השני מייצג את הסכום שהעביר התלמיד לקופת הכיתה. הפלט של האלגוריתם הוא הסכום המצטבר של כל אחת מהכיתות וכן הסכום הכולל שהצטבר בשכבה.

בעיה 2: בחנות הנעליים "נעל לכל" מעוניינים לדעת מהו הפדיון היומי מסך כל המכירות בכל אחת מ-5 המחלקות בחנות (המחלקות ממוספרות מ-1 עד 5). כתבו אלגוריתם שהקלט שלו הוא סדרה של זוגות מספרים, כאשר המספר הראשון בכל זוג מייצג את מספר המחלקה והמספר השני מייצג את המחיר של סכום הקניה של לקוח. סוף הקלט יסומן על ידי זוג שהמספר הראשון בו הוא 0. הפלט של האלגוריתם הוא התפלגות המכירות לפי מחלקות (כלומר, עבור כל מחלקה הפדיון היומי הכולל שלה).

עבור הפתרונות של שתי הבעיות האלגוריתמיות דרושים לנו כמה צוברים. בבעיה 1 אנו צריכים שבעה צוברים עבור הסכומים המצטברים לכל אחת מהכיתות, ובבעיה 2 אנו צריכים חמישה צוברים (צובר סכום מכירות לכל מחלקה). כדי להימנע מסירבול, נגדיר מערך שגודלו כמספר הצוברים הנדרש (7 עבור בעיה 1 ו-5 עבור בעיה 2). בבעיה 1 מצייני המערך מייצגים את מספרי הכיתות, ובבעיה 2 הם מייצגים את מספרי המחלקות. כל תא במערך מייצג צובר המתאים למציון. זהו אלגוריתם המתבסס על התבנית **מעריך צוברים**. הרעיון בבסיסה של התבנית **מעריך צוברים** דומה לתבנית של מערך המונים. ההבדל הוא באופי הפעולה על כל תא במערך: מנייה במערך המונים לעומת צבירה במערך הצוברים.

נתבונן בשני האלגוריתמים הבאים, הראשון לפתרון בעיה 1 והשני לפתרון בעיה 2:

1. כצט 247 פסאים:

```
1.1 קלוט מספר כינה classNum-2 כספ שהכווו גלמיז כ-  
sumStudent
```

```
1.2 הפצא אג sumClasses[classNum-1]-2 sumStudent
```

2. אגף אגף sumTotal-0

3. עבור כל i שלם בגוון 0 עד 6 בצד:

3.1 הצג כפוף: הסכום המצטבר לכיתה $i+1$ הוא: $\text{sumClasses}[i]$

3.2 הצג אגף sumTotal ב- $\text{sumClasses}[i]$

4. הצג כפוף: הסכום המצטבר לשכבה הוא sumTotal

1. קאוס מספר מחלקה ב- departmentNum וסכום הקניה של לקוח ב- price

2. כל ערך departmentNum שונה מ-0 בצד:

2.1 הצג אגף $\text{sumDepartments}[\text{departmentNum}-1]$ ב- price

2.2 קאוס מספר מחלקה ב- departmentNum וסכום הקניה של לקוח ב- price

3. עבור כל i שלם בגוון 0 עד 4 בצד:

3.1 הצג כפוף: הפדיון היומי למחלקה $i+1$ הוא: $\text{sumDepartments}[i]$

בשני קטעי האלגוריתמים האלה ניתן לזהות תבנית **צבירה**, אלא שהיא מופעלת על כמה צוברים הפועלים במקביל, ושמורים במערך אחד.

נציג את מאפייני התבנית **מערך צוברים**, עבור סדרת ערכים שאורכה ידוע מראש. ניתן להתאים את מאפייני התבנית למקרים בהם משך הביצוע תלוי בתנאי, בדומה לפתרון בעיה 1. התבנית מתייחסת לצבירה על ידי סכום, אך ניתן להציג מאפיינים של תבנית דומה המתייחסת לצבירה על ידי מכפלה.

שם התבנית: מערך צוברים

נקודת מוצא: אורך סדרת הערכים limit, סדרת ערכים, ביטוי חשבוני whichSum המקשר בין ערך בסדרה למציון של המערך

מטרה: בניית מערך צוברים עבור ערכי הסדרה, שאורכה limit

אלגוריתם:

1. אגף את ערכי מערך הצוברים

2. כצב limit פעמים:

2.1. השם את הערך הכא כסדרה element-

2.2. השם את הערך לצבירה כ-value

2.3. הוסף ל-sumElements[whichSum] את ערכו value

יישום ב-C#:

```
for (i = 1; i <= limit; i++)
{
    element = הבא בסדרה;
    value = הערך לצבירה;
    sumElements[whichSum] += value;
}
```

שימו ♥ : גם התבנית של **מערך צוברים**, הוצגה באופן כללי, כך שהיא מתאימה למגוון בעיות :

♦ לא נקבעו במפורש הערכים לאתחול. במקרה הפשוט, מאותחלים כל הצוברים ל-0. במקרה כזה, ביישום בשפת C# אין צורך לבצע אתחול מפורש, משום שב-C# מתבצע אתחול אוטומטי של איברי מערך שלמים ל-0.

♦ בדומה לתבנית **מערך מונים**, גם התבנית של **מערך צוברים** אינה מפרטת מהיכן מגיעים איברי הסדרה. ערכים אלה יכולים, למשל, להתקבל כקלט או מקריאת ערכי מערך אחר.

♦ התבנית של **מערך צוברים** גם אינה מפרטת כצד מחושבים הערכים לצבירה. גם ערכים אלה יכולים, למשל, להתקבל כקלט או מקריאת ערכי מערך אחר, וייתכן כי ניתן לחשב, עבור כל איבר element, בעזרת ביטוי חשבוני מסוים תלוי ב-element, את ערך הצבירה המתאים לו value.

♦ כמו בתבנית **מערך מונים**, גם בתבנית של **מערך צוברים** המשתנה element מתייחס לאיבר התורן בסדרה, ולמרות שאין זה מופיע במפורש באלגוריתם שניתן עבור התבנית וביישום שלו, הרי פעולת הצבירה תלויה ב-element: הביטוי whichSum תלוי ב-element ומקשר בין ערכו לערך של מציון במערך.

שאלה 4

ישמו את שני האלגוריתמים עבור בעיות 1 ו-2 בשפת C#.

שאלה 5

לאור הצלחת המבצע לסיוע למשפחות נזקקות החליטה הנהלת בית-הספר "נוף ים" להעניק יום טיול לנגב לתלמידי הכיתה שאספה את הסכום המירבי. יש להרחיב את האלגוריתם כך שיוצג כפלט גם מספר הכיתה שזכתה ביום הטיול. אם יש יותר מכיתה אחת שאספה את הסכום המירבי יוענק יום טיול לכולן.

נתון האלגוריתם הבא לפתרון הבעיה:

1. אגף אג איברי המערך sumClasses ל-0

2. כצט 247 קסמית:

2.1. קאט מספר כינה ב-classNum וסכום כספ שהוויג אמיז ב-

sumStudent

2.2. הגצא אג sumClasses[classNum-1] ב-sumStudent

3. אגף אג sumTotal ל-0

4. עכור כלי i שלם בגומ 0 ע 6 כצט:

4.1. הגצ כפוט: הסכום המצטבר לכינה i+1 הוא: sumClasses[i]

4.2. הגצא אג sumTotal ב-sumClasses[i]

5. הגצ כפוט: הסכום המצטבר לשכבה הוא sumTotal

6. אגף אג max ל-sumClasses[0]

7. עכור כלי i שלם בגומ 0 ע 6 כצט:

7.1. אס sumClasses[i] > max

7.1.1. השס ב-max אג הערך של sumClasses[i]

8. עכור כלי i שלם בגומ 0 ע 6 כצט:

8.1. אס ערכו של sumClasses[i] שווה ל-max

8.1.1. הגצ כפוט: כינה מספר i+1 זוכה ביום טיול אלזב

א. ציינו מהן התבניות המשולבות באלגוריתם.

ב. ישמו את האלגוריתם בשפת C# (כזכור, אין צורך ליישם את שלב אתחול המערך).

שאלה 6

בחנות הנעליים "נעל לכל" מעוניינים לדעת את הפדיון מסך כל המכירות בכל אחד מימות השבוע. יש לכתוב אלגוריתם שהקלט שלו הוא סדרות של ספרים ממשיים (המסתיימות בזקיף 0) – סדרה

עבור כל יום מהימים א' עד ו'. המספרים הממשיים מייצגים את סכומי הקניות של הלקוחות. הפלט של האלגוריתם הוא התפלגות המכירות לפי ימים (כלומר, עבור כל יום הפדיון הכולל שלו). נתון האלגוריתם הבא לפתרון הבעיה:

1. $0 \leq \text{sumDays}$ איברי המערך

2. עבור כל day שלם בגוון $0 \leq \text{day} < 5$:

2.1 קאוס סכום קניה price

2.2 כל price של $0 \leq \text{price} < 5$:

2.2.1 הגדל $\text{sumDays}[\text{day}]$

2.2.2 קאוס סכום קניה price

3. עבור כל day שלם בגוון $0 \leq \text{day} < 5$:

3.1 הגדל כפאוס: הפדיון היומי עבור יום $\text{day}+1$ הוא $\text{sumDays}[\text{day}]$

- א. באלגוריתם ישנו שימוש בתבנית מערך צוברים. הציעו אלגוריתם השקול לאלגוריתם הנתון ללא שימוש בתבנית מערך צוברים. ציינו באילו תבניות השתמשתם עבור כתיבת האלגוריתם.
- ב. ישמו את האלגוריתם הנתון בשפת C# (כזכור, אין צורך ליישם את שלב אתחול המערך).

שאלה 7

א. במוסד לביטוח רפואי בוצע מחקר על צריכת 150 תרופות על ידי החולים המבוטחים. לכל תרופה מספר סידורי בין 1 ל-150. פתחו אלגוריתם, שהקלט שלו הוא סדרת שלשות של ערכים, כאשר כל שלשה מייצגת גיל של מבוטח, מספר סידורי של התרופה שנצרכה והכמות שלה (מספר יחידות). סדרת הקלט מסתיימת עם קליטת הזקיף 1- כגיל המבוטח. הפלט של האלגוריתם הוא קבוצת הגילאים הצורכת כמות תרופות כוללת גדולה ביותר, מבין 4 קבוצות הגילאים הבאות: קבוצת גילאי 0 עד 10, קבוצת גילאי 11 עד 30, קבוצת גילאי 31 עד 50, קבוצת גילאי 51 ומעלה. ייתכן שיותר מקבוצת גיל אחת צורכת את כמות התרופות הכוללת הגדולה ביותר.

שימו לב: לצורך פתרון סעיף זה אין צורך להבחין בין סוגי התרופות השונים.

ב. הרחיבו את האלגוריתם כך שיוצגו כפלט גם מספרי התרופות שלא נצרכו כלל.

ג. ציינו באילו תבניות השתמשתם בכתיבת האלגוריתם וכיצד שילבתם ביניהן.

ד. ישמו את האלגוריתם בשפת C#.

חישוב שכיח

נתבונן בבעיה האלגוריתמית הבאה:

כתבו אלגוריתם שהקלט שלו הוא 58 ספרות והפלט שלו הוא הספרה שהופיעה בקלט הכי הרבה פעמים. אם יש יותר מספרה אחת כזאת, יש להציג את כולן.

לצורך פתרון הבעיה צריך קודם כל למנות את מספר המופעים של כל אחת מהספרות. לשם כך, נשתמש בתבנית **מערך מונים**. לאחר מכן, עלינו לחשב את הערך המקסימלי מבין ערכי המונים. זהו למעשה הערך השכיח, כלומר הערך שהופיע מספר פעמים רב ביותר. לצורך כך נוכל להשתמש בתבנית של **מציאת מקסימום בסדרה**, כדי למצוא את הערך המקסימלי, ולאחר מכן להשתמש בתבנית של **מציאת כל הערכים בסדרה המקיימים תנאי**, כאשר התנאי הוא שוויון לערך המקסימלי, תוך הצגה כפלט של מצייני הערכים שנמצאו. יש לשים לב, שעלינו לבצע שינויים קלים בתבניות **מציאת מקסימום בסדרה** ו-**מציאת כל הערכים בסדרה המקיימים תנאי** מאחר שהערכים בסדרה אינם נקראים מהקלט, אלא שמורים במעריך **מונים**.

הנה האלגוריתם המלא:

1. **כצע 58 פעמים:**

1.1 קלוט ספרה $digit$

1.2 הציף את $countDigitsArr[digit]$ ב-1

2. השם max את $countDigitsArr[0]$

3. עבור כל i שלם בגוומ $0 \leq i < 10$: **צע**

3.1 אם $countDigitsArr[i] > max$

3.1.1 השם max את הערך של $countDigitsArr[i]$

4. עבור כל i שלם בגוומ $0 \leq i < 10$: **צע**

4.1 אם $countDigitsArr[i] == max$

4.1.1 הציף כפלוט: ספרה i הופיעה הכי הרבה פעמים

נציג את מאפייני התבנית **חישוב שכיח** עבור ביצוע חוזר שאורכו ידוע מראש. ניתן להתאים את מאפייני התבנית לביצוע חוזר התלוי בתנאי.

שם התבנית: חישוב שכיח

נקודת מוצא: אורך סדרת הערכים limit, סדרת ערכים, ביטוי חשבוני whichSum המקשר בין ערך בסדרה למציין של המערך

מטרה: הצגה כפלט של הערך השכיח או של הערכים השכיחים בסדרת הקלט, שאורכה הוא

limit

אלגוריתם:

1. אגף את ערכי המערך המונים

2. בצע limit פעמים:

2.1. השם את הערך הבא בסדרה ב-element

2.2. הגדל את sumElements[whichCount] ב-1

3. אגף את max בערכו של countElements[0]

4. עבור כל i שלם במגוון מ-1 עד אורך המערך countElements פגום ו בצע:

4.1. אם $countElements[i] > max$

4.1.1. השם ב-max את הערך של countElements[i]

5. עבור כל i שלם במגוון מ-1 עד אורך המערך countElements פגום ו בצע:

5.1. אסערכו של countElements[i] שווה ל-max

5.1.1. הצג את i כפלט

יישום ב-C#:

```
for (i = 1; i <= limit; i++)
{
    element = הבא בסדרה;
    sumElements[whichCount]++;
}
max = countElements[0];
for (i = 1; i < countElements.length; i++)
{
    if (countElements[i] > max)
    {
        max = countElements[i];
    }
}
for (i = 0; i < countElements.length; i++)
{
    if (countElements[i] == max)
    {
        Console.WriteLine(counts[i]);
    }
}
```

שימו ♥ : באלגוריתם של התבנית בחרנו להציג כפלט את הערך השכיח או הערכים השכיחים אבל ניתן לבצע על ערכים אלו פעולות חישוביות שונות כגון מנייה, צבירה וכו'.

שאלה 8

המורה אמיר החליט לבצע כמה עיבודים סטטיסטיים על ציוני 40 תלמידיו ב"יסודות מדעי המחשב". אמיר חילק את ציוני התלמידים, הנעים בין 0 ל-100, ל-10 קבוצות באופן הבא: קבוצת הציונים בין 0 ל-10 (קבוצת ציונים ראשונה), קבוצת הציונים בין 11 ל-20 (קבוצת ציונים שנייה) וכך הלאה עד קבוצת הציונים בין 91 ל-100 (קבוצת ציונים עשירית).

א. כתבו אלגוריתם, שהקלט שלו הוא ציוני 40 התלמידים והפלט שלו הוא מספר התלמידים בכל קבוצת ציונים, וכן קבוצת הציונים השכיחה (תיתכן יותר מקבוצה אחת).

ב. הרחיבו את האלגוריתם שכתבתם בסעיף א כך שיציג כפלט את טווח הציונים, כלומר, ההפרש בין הציון הגבוה ביותר במבחן לבין הציון הנמוך ביותר במבחן. ציינו באילו תבניות נוספות השתמשתם וכיצד **שילבתם** ביניהן.

ג. הרחיבו את האלגוריתם שכתבתם בסעיף ב כך שיציג כפלט הודעה האם הציונים מתפלגים סימטרית, כלומר, מספר התלמידים בקבוצה הראשונה שווה למספר התלמידים בקבוצה העשירית, מספר התלמידים בקבוצה השנייה שווה למספר התלמידים בקבוצה התשיעית וכן הלאה. ציינו באילו תבניות נוספות השתמשתם וכיצד **שילבתם** ביניהן.

ד. הרחיבו את האלגוריתם שכתבתם בסעיף ג כך שאם הציונים מתפלגים סימטרית, על האלגוריתם להציג הודעה האם הציונים מתפלגים סימטרית בצורת פעמון, כלומר, הערכים ב-5 קבוצות הציונים הראשונות מהווים סדרה עולה ממש, והערכים ב-5 קבוצות הציונים האחרונות מהווים סדרה יורדת ממש. ציינו באילו תבניות נוספות השתמשתם וכיצד **שילבתם** ביניהן.

ה. ישמו את האלגוריתם המורחב שכתבתם בסעיף ד בשפת C#.

הזזה מעגלית בסדרה

בפרק 3 הראינו את התבנית **הזזה מעגלית בסדרה** עבור סדרה בת שני ערכים. עתה נרחיב את התבנית עבור סדרת ערכים באורך כלשהו. נזכיר כי ישנם שני סוגי הזזות מעגליות: הזזה מעגלית שמאלה והזזה מעגלית ימינה.

נפריד את מאפייני התבנית **הזזה מעגלית בסדרה** לשתי תת-תבניות: ראשית נציג את מאפייני התבנית **הזזה מעגלית שמאלה בסדרה** ואחר כך נציג את מאפייני התבנית **הזזה מעגלית ימינה בסדרה**.

שם התבנית: הזזה מעגלית שמאלה בסדרה

נקודת מוצא: סדרת ערכים במערך elements שאורכו length

מטרה: הזזה מעגלית שמאלה של ערכי המערך

אלגוריתם:

1. השם temp-2 אג elements[0]

2. עבור כל i אגם בגומס 0 עד length-2 בצע:

2.1. השם ב-elements[i] אג העוק א elements[i+1]

3. השם ב-elements[length-1] אג העוק א temp

יישום ב-C#:

```
temp = elements[0];
for (i = 0; i <= elements.length - 2; i++)
{
    elements[i] = elements[i + 1];
}
elements[elements.length - 1] = temp;
```

שם התבנית: הזזה מעגלית ימינה בסדרה

נקודת מוצא: סדרת ערכים במערך elements שאורכו length

מטרה: הזזה מעגלית ימינה של ערכי המערך

אלגוריתם:

1. השם temp-2 אג elements[length-1]

2. עבור כל i אגם בגומס length-1 עד 1 (כסדר יורד) כצט:

2.1 השם כ-elements[i] אג הערך א elements[i-1]

3. השם כ-elements[0] אג הערך א temp

יישום ב-C#:

```
temp = elements[elements.length - 1];
for (i = elements.length - 1; i >= 1; i--)
{
    elements[i] = elements[i - 1];
}
elements[0] = temp;
```

שימו ♥: לשם פשטות, התבנית מניחה שאיברי הסדרה המיועדים להזזה נמצאים במערך elements, האחד אחרי השני. אבל, ייתכן שהסדרה שבה נדרש לבצע הזזה מעגלית היא תת-סדרה (רצופה או לא רצופה) של איברי המערך. שאלה 10 מתייחסת לבעיה כזאת.

שאלה 9

נתון אלגוריתם, שהקלט שלו הוא 6 מספרים שיישמרו במערך numbers, ומספר שלם חיובי num:

1. עבור כל i אגם בגומס 1 עד 6 כצט:

1.1 קאוט מספר אגם כ-numbers[i]

2. קאוט מספר אגם ג'ויב כ-num

3. כצט num פסמיס:

3.1 הזזה מעגלית שמאלה במערך numbers

א. נקלטו הערכים הבאים למערך numbers: 21 34 9 78 6 4

1. מה יהיה הפלט של האלגוריתם עבור הקלט 5 ל-num?

2. מה יהיה הפלט של האלגוריתם עבור הקלט 35 ל-num?

- ב. מהו מספר הפעמים שמתבצעת פעולת התבנית: **הזזה מעגלית שמאלה**?
- ג. כתבו אלגוריתם **יעיל** יותר, השקול לאלגוריתם הנתון כך שפעולת התבנית: **הזזה מעגלית שמאלה** תתבצע מספר קטן יותר של פעמים. ציינו באיזו תבנית השתמשתם עבור כתיבת האלגוריתם **היעיל**.
- ד. ישמו את האלגוריתם **היעיל** בשפת C#.

שאלה 10

- א. כתבו אלגוריתם, שהקלט שלו הוא 16 מספרים למערך, והפלט שלו הוא ערכי המערך לאחר הזזה מעגלית שמאלה עבור סדרת המספרים הנמצאים במקומות האי-זוגיים במערך והזזה מעגלית ימינה עבור סדרת המספרים הנמצאים במקומות הזוגיים במערך.
- ב. ישמו את האלגוריתם בשפת C#.
-

הזזה של תת-סדרה

הזזה של תת-סדרה היא תבנית הנחוצה בהקשרים רבים בעיבוד סדרות. **הזזה של תת-סדרה** היא תבנית הזזה ליניארית (לא מעגלית) של תת-סדרה של ערכים אל מקום אחד **שמאלה** או אל מקום אחד **ימינה**. הנה שתי דוגמאות בולטות לשימוש בתבנית: הוצאת ערך ממקום k בסדרה ו"צמצומה לשמאל", כלומר, הזזה במקום אחד שמאלה של כל האיברים מהמקום $k+1$ וימינה; "ריווח הסדרה ימינה" על ידי הזזה של הערכים החל מהמקום $k+1$ ימינה כדי לפנות מקום להכנסה של איבר חדש לסדרה במקום k . במקרה של "צמצום לשמאל" מתבצעת פעולת "הוצאה" של ערך מהמקום k לפני ההזזה, ובמקרה של "ריווח ימינה" תתבצע פעולת "הכנסה" של ערך חדש למקום k אחרי ההזזה. באופן דומה ניתן לבצע "צמצום לימין" ו-"ריווח שמאלה".

נפריד את מאפייני התבנית **הזזה של תת-סדרה** לשתי תת-תבניות: ראשית נציג את מאפייני התבנית **הזזה של תת-סדרה שמאלה** ואחר כך נציג את מאפייני התבנית **הזזה של תת-סדרה ימינה**.

שם התבנית: הזזה של תת-סדרה שמאלה

נקודת מוצא: סדרת ערכים במערך elements באורך length, מקום k במערך $(0 \leq k < \text{length}-1)$

מטרה: הזזה שמאלה של התת-סדרה הנמצאת במקומות $k+1..length-1$ למקומות $k..length-2$

אלגוריתם:

1. עבור כל i שלם בגוון k עד $length-2$ בצד:

1.1 השם $elements[i]$ אל השם $elements[i+1]$

יישום ב-C#:

```
for (i = k; i <= elements.length - 2; i++)
{
    elements[i] = elements[i + 1];
}
```

שם התבנית: הזזה של תת-סדרה ימינה

נקודת מוצא: סדרת ערכים במערך elements באורך length, מקום k במערך $(0 < k \leq \text{length}-1)$

מטרה: הזזה ימינה של התת-סדרה הנמצאת במקומות $k-1 \dots 0$ למקומות $k \dots 1$

אלגוריתם:

1. עבור כל i אלמנט באורך $k-1$ (כנסד יורד) כ33:

1.1. העם i elements[i] אל העם $i-1$ elements[i-1]

יישום ב-C#:

```
for (i = k; i >= 1; i--)  
{  
    elements[i] = elements[i - 1];  
}
```

שאלה 11

ברשימת החולים המוזמנים לרופא מומחה נקבעה לכל מוזמן פגישה, החל מהשעה 16:00 ועד השעה 20:00, כאשר לכל מוזמן מוקדש פרק זמן של חצי שעה.

מקרה בהול גרם להכנסת חולה חדש לרשימה בשעה 18:30, ולכן יש לעדכן אצל המזכירה הרפואית את רשימת המוזמנים, המסודרת לפי סדר פגישתם המיועדת עם הרופא.

א. כתבו אלגוריתם, שהקלט שלו הוא הרשימה התחילית של מספרי הזהות של 9 החולים המוזמנים וכן את מספר הזהות של החולה החדש, והפלט שלו הוא רשימת מספרי הזהות של החולים, על פי הסדר לאחר העדכון, כאשר לכל חולה מוצגת גם שעת הפגישה המיועדת לו עם הרופא.

ב. ישמו את האלגוריתם בשפת C#.

שאלה 12

ב"מכרז המבטיח" מנהלים רישום של הצעות רכישה ל-5 מערכות ישיבה לסלון. המידע נשמר עבור 20 ההצעות הגבוהות ביותר והן מסודרות לפי סדר יורד.

א. כתבו אלגוריתם, שהקלט שלו הוא 20 הסכומים של ההצעות הגבוהות שהתקבלו עד כה, וכן סכום של הצעה חדשה, והפלט שלו הוא 20 ההצעות הגבוהות ביותר לאחר העדכון של ההצעה החדשה.

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

ג. הרחיבו את האלגוריתם כך שיציג כפלט את הסכומים של הזוכים ב-5 מערכות הישיבה לסלון. ציינו באיזו תבנית נוספת השתמשתם וכיצד **שילבתם** אותה באלגוריתם.

היפוך סדר האיברים בסדרה

בפרק 3 הראינו את התבנית **היפוך סדר האיברים בסדרה** עבור סדרה בת שני איברים. עתה נרחיב את התבנית עבור סדרת איברים באורך כלשהו.

לצורך **היפוך סדר הערכים בסדרה** אפשר להחליף בין האיבר הראשון בסדרה לבין האיבר האחרון בסדרה, להחליף בין האיבר השני בסדרה לבין האיבר הלפני אחרון בסדרה וכן הלאה.

נציג את מאפייני התבנית **היפוך סדר האיברים בסדרה**:

שם התבנית: היפוך סדר האיברים בסדרה

נקודת מוצא: סדרת ערכים במערך elements

מטרה: היפוך סדר האיברים במערך, שאורכו length

אלגוריתם:

1. השם limit-2 אגמנת החלוקה של length פריטים לשת קבוצות

2. עבור כל i אשם בגומא מ-0 עד limit-1 כ3ע:

2.1 החלף את הערכים של elements[i] ושל elements[length-i-1]

שימו ♥: ההחלפה מתבצעת עד מחצית אורך סדרת האיברים. אם אורכה אינו זוגי, אז האיבר האמצעי אינו מוחלף עם אף איבר, ונשאר במקומו, כפי שאכן צריך להיות.

שאלה 13

א. כתבו אלגוריתם, שהקלט שלו הוא 15 מספרים למערך, המסודרים בסדר עולה, והפלט שלו הוא ערכי המערך לאחר היפוכם.

ב. שנו את האלגוריתם שכתבתם בסעיף א כך שיהפוך את הסדר של 12 האיברים הראשונים במערך. שלושת הערכים הגדולים ביותר במערך יישארו במקומם. הפלט יהיה ערכי המערך לאחר ההיפוך.

ג. ניתן לבצע את האלגוריתמים שבסעיפים א ו-ב **ללא** שימוש בתבנית **היפוך סדר האיברים בסדרה**. כתבו את האלגוריתמים המתאימים.