

פתרון מבחן בגרות תש"ס 2000

פתרון 2000

פתרון סדרק רא"ל - 50 נקודות

פתרון שאלה 11

א. טבלת מעקב אחר ביצוע האלגוריתם הנתון עבור הערכים $num = 37, sif = 7$

שורה	num	sif	האם ספרת האחדות של num שווה ל-sif	r	פלט
(1)-(2)	37	7			
(3)			אמת ($7 = 7$)	37	
(4)					37

יודפס: 37.

ב. עבור הערכים: $num = 52, sif = 2$, תתבצע שורת האלגוריתם (3.1) ויודפס 52.

הסבר הפתרון עבור סעיף ב

שורה	num	sif	האם ספרת האחדות של num שווה ל-sif	r	פלט
(1)-(2)	52	2			
(3)			אמת ($2 = 2$)	52	
(4)					52



כדי שתתבצע שורת האלגוריתם (3.1) התנאי צריך להתקיים, כלומר ספרת האחדות של num צריכה להיות זהה ל-sif. מאחר וספרת האחדות של 52 היא 2 התנאי מתקיים.

ג. עבור הערכים: $num = 52, sif = 1$, תתבצע שורת האלגוריתם (3.2) ויודפס 3.

הסבר הפתרון עבור סעיף ג:

שורה	num	sif	האם ספרת האחדות של num שווה ל-sif	r	פלט
(1)-(2)	52	1			
(3)			שקר ($2 = 1$)	$3 = (2 + 1)$	
(4)					3



כדי שתתבצע שורת האלגוריתם (3.2) התנאי לא צריך להתקיים, כלומר ספרת האחדות של num צריכה להיות שונה מ-sif. מאחר וספרת האחדות של 52 היא 2 התנאי לא מתקיים.

פתרון שאלה 12

א. טבלת מעקב אחר ביצוע האלגוריתם הנתון עבור הערכים $x=6, y=3$

שורה	pull	x	y	$x > y$	פלט
(1)-(2)-(3)	0	6	3		
(4)-(4.1)-(4.2)	4		4	אמת ($6 > 3$)	
(4)-(4.1)-(4.2)	9		5	אמת ($6 > 4$)	
(4)-(4.1)-(4.2)	15		6	אמת ($6 > 5$)	
(4)				שקר ($6 > 6$)	
(5)					15 6 6

יודפס: 15 6 6.

ב. הלולאה לא תתבצע כלל עבור הערכים: $x = 6, y = 8$. במקרה זה יודפס: 0 8 6.

הסבר הפתרון

טבלת מעקב אחר ביצוע האלגוריתם הנתון עבור הערכים $x = 6, y = 8$



שורה	pull	x	y	$x > y$	פלט
(1)-(2)-(3)	0	6	8		
(4)				שקר ($6 > 8$)	
(5)					0 8 6

יודפס: 0 8 6.

כדי שהלולאה לא תתבצע כלל צריך תנאי הלולאה ב-(4) לא להתקיים כבר בפעם הראשונה לכן הקלט צריך להיות כזה כך ש- $x \leq y$. במקרה זה pull לא מתעדכן כלל ונשאר 0, וכן המשתנים x ו- y נשארים זהים לקלט.

פתרון שאלה 13

טבלת מעקב עבור האלגוריתם של $n = 3$

שורה	n	i	b	פלט
(1)	3			
(2)-(2.1)-(2.2)		1	2	2
(2)-(2.1)-(2.2)		2	4	4
(2)-(2.1)-(2.2)		3	6	6

טבלת מעקב עבור האלגוריתם של $n = 2$

שורה	n	i	b	פלט
(1)	2			
(2)-(2.1)-(2.2)		1	2	2
(2)-(2.1)-(2.2)		2	4	4

האלגוריתם של n מבצע את הנדרש. עבור כל ערך של n , הלולאה מתבצעת n פעמים, כלומר יודפסו n ערכים כפי שצריך. b המודפס למעשה גדל תמיד ב-2.

טבלת מעקב אחר ביצוע האלגוריתם של **לילך** עבור: $n = 2$

שורה	n	x	$x < n * 2$	פלט
(1)-(2)	2	0		
(3)-(3.1)-(3.2)		2	אמת ($0 < 2 * 2$)	2
(3)-(3.1)-(3.2)		4	אמת ($2 < 2 * 2$)	4
(3)			שקר ($4 < 2 * 2$)	

טבלת מעקב אחר ביצוע האלגוריתם של **לילך** עבור: $n = 3$

שורה	n	x	$x < n * 2$	פלט
(1)-(2)	3	0		
(3)-(3.1)-(3.2)		2	אמת ($0 < 3 * 2$)	2
(3)-(3.1)-(3.2)		4	אמת ($2 < 3 * 2$)	4
(3)-(3.1)-(3.2)		6	שקר ($4 < 3 * 2$)	6
(3)			שקר ($6 < 3 * 2$)	

האלגוריתם של **לילך** מבצע את הנדרש. עבור כל ערך של n , הלולאה מתבצעת פעמים, כלומר יודפסו ערכים כפי שצריך. x המודפס גדל תמיד ב-2 ואז מודפס. בפעם האחרונה שהלולאה מתבצעת $x < 2 * n$, אך גדל בפנים בדיוק לערך האחרון הנכון $2 * n$ ומודפס. על פי דרישות השאלה מספיקה טבלת מעקב אחת עבור כל אלגוריתם מאחר ועל פי מטרת האלגוריתם יתכן שיהיה הבדל בין זוגי ל- n אי-זוגי, מוטב לבדוק עבור קלטים כאלו



פתרון שאלה 14

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	מערך חד מימדי	אחסון נתונים נדרשים	שלמים
i	שלם	מצוין במערך a	0..99

טבלת תת-משימות

סוג הפעולה	מטרת תת-המשימה
לולאה	השמת ערכים אקראיים בתחום 1-20 בתאי המערך
לולאה	הדפסת איברי המערך מן הראשון עד לאחרון
לולאה	הדפסת איברי המערך מן האחרון ועד הראשון

אלגוריתם לפתרון הבעיה

- (1) עבור i מ-0 עד 99 בצע
 - (1.1) צור מספר אקראי בתחום בין 1 ל-20 והשם אותו ב- $[i]$
 - (2) הצג כפלט 'אברי המערך מן הראשון לאחרון'
 - (3) עבור i מ-0 עד 99 בצע
 - (1.3) הצג כפלט את $[i]$
 - (4) הצג כפלט 'אברי המערך מן האחרון לראשון'
 - (5) עבור i מ-99 עד 0 בצע
 - (1.5) הצג כנ"ל

מפתח פתרונות



מימוש האלגוריתם

// q14t00.c

```
using System;
public static void
Main()
{
    int a = new int[100];
    int i;
    Random rnd = new Random();
    for ( i = 0; i < a.Length; i++)
        a [i] = rnd.Next(1, 21);
    Console.WriteLine ("The array elements from first to last : \n");
    for ( i = 0; i <a.Length; i++)
        Console.Write (a[i] + " ");
    Console.WriteLine ("\nThe array elements from last to first : \n");
    for ( i = a.Length - 1; i >= 0; i--)
        Console.Write (a[i] + " ");
}
```

/* קלט : הגדרת מערך חד-מימדי בגודל 100
/* פלט : התכנית משימה ערכים אקראיים בתום 20.1 במעריך,
/* ומדפיסה אותו לפי הסדר מראשון לאחרון ומאחרון לראשון*/

/* לולאה לבניית המערך*/

/* הגרלת ערך שלם בתחום 1..20 והשמתו במערך*/

/* הדפסת המערך מראשון עד אחרון*/

/* הדפסת המערך מאחרון עד ראשון*/



- ✓ יש לייצר מספרים אקראיים שלמים בת חום 1..20. הפעולה Random.Next(x, y) מחזירה ערכים שלמים מ-x (כולל) ועד y (לא כולל).
- ✓ לא משתמשים פעמים רבות בלולאה בסדר יורד. הלולאה פועלת בדיוק כמו כל לולאת for רגילה, אלא שבכל פעם קטן ערכו של i ב-1.

פתרון נוסף : ניתן לכתוב את הלולאה היורדת גם ללא שימוש בלו לאת for עם מציון יורד, אלא על ידי שימוש בלולאת while עם מונה ידני יורד:

```
i = a.Length - 1;
while ( i >= 0)
{
    Console.Write (a[i] + " ");
    i --;
}
```

/* הדפסת המערך מאחרון עד ראשון*/



בשאלה בה יש דרישה לפיתוח ויישום של אלגוריתם יש לפעול על פי השלבים הבאים:

- (א) חלוקה לתת-משימות.
 - (ב) הגדרת משתנים עיקריים.
 - (ג) פיתוח אלגוריתמים לתת-המשימות.
 - (ד) מימוש האלגוריתם על ידי תכנית השלבים (א) – (ב) יכולים להיות חלופיים
- בדרך כלל אין דרישה לכתוב אלגוריתם וגם תכנית- אך אם כתוב פתח ויישם ללא הקלות יש לכתוב את שניהם. בבעיות מורכבות המופיעות בדרך כלל בחלק ג של הבחינה – תמומש כל תת-משימה על ידי פונקציה. בשאלות קטנות אין צורך להגדיר תת- תכניות לכל תת- משימה כי הן מסרבלות את הפתרון (אך ניתן לעשות זאת).

פתרון שאלה 15

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
num1	שלם	מספר ראשון בזוג המשווה	שלמים
num2	שלם	מספר שני בזוג המשווה – קלט אחרי num1	שלמים
i	שלם	מונה לולאה – מונה קלטים	1..99

טבלת תת-משימות

מטרת תת-המשימה	סוג הפעולה
קליטת האיבר הבא השווה לקודמנו אם הוא גדול יותר להדפיסן החלפת סדר המשתנים	לולאה

אלגוריתם לפתרון הבעיה

- (1) קלוט מספר למשתנה num1
- (2) עבור i מ-2 עד 100 בצע
 - (1.2) קלוט מספר למשתנה num2
 - (2.2) אם $num2 > num1$ אזי
 - (1.2.2) הצג כפלט את num2
 - (3.2) השם את num2 ב- num1

מימוש האלגוריתם

// q15t00.c

```

/* קלט: 100 מספרים שלמים
/* פלט: מספרי הקלט הגדולים מן המספר שקלט לפניהם */
public static void Main()
{
    int i, num1, num2;
    num1 = int.Parse(Console.ReadLine()); /* קלט נתון ראשון */
    for ( i = 1; i < 100; i++ ) /* לולאה לקליטת שאר 99 הנתונים */
    {
        num2 = int.Parse(Console.ReadLine()); /* קלט נתון שני */
        if ( num2 > num1 ) /* אם הקלט האחרון גדול מקודמו */
            Console.WriteLine (" --> " + num2 + "\n"); /* אזי הדפס את הקלט האחרון */
        num1 = num2; /* "גלגלי" את הנתונים – האחרון הופך להיות "לפני אחרון" */
    }
}

```

✓ מאחר והנתון הראשון נקלט מחוץ ללולאה, יש לבצע את הלולאה פעם אחת פחות בכדי לקלוט 100 נתונים.

תבנית "גילגולי" המשתנים חוזרת על עצמה בבעיות רבות. כדי להשוות בין כל זוג של איברים, בכל פעם מעבירים את "האחרון" להיות "הלפני אחרון" וקולטים "אחרון" חדש. ראה במעקב בטבלת המעקב הבאה עבור קלט של 6 נתונים בלבד משמאל לימין: 1, 7, 4, 6, 7, 3.



שורה	i	num1	num2	num2 > num1	פלט
------	---	------	------	-------------	-----

(1)		3			
(2)-(2.1)-(2.2)	2		7	אמת ($7 > 3$)	7
(2.3)		7			
(2)-(2.1)-(2.2)	3		4	שקר ($4 > 7$)	
(2.3)		4			
(2)-(2.1)-(2.2)	4		6	אמת ($6 > 4$)	6
(2.3)		6			
(2)-(2.1)-(2.2)	5		7	אמת ($7 > 6$)	7
(2.3)		7			
(2)-(2.1)-(2.2)	6		1	שקר ($1 > 7$)	
(2.3)		1			

שים ♥ : "גילגול" הנתונים חייב להתבצע בכל מקרה לכן אסור שיהיה בתוך התנאי

ביצוע קלט של זוג נתונים בתוך הלולאה והשוואה ביניהם. במקרה כזה ההשוואה היא רק בין צמדים ולא בין כל הזוגות בקלט. כל נתון צריך להיות מושווה לקודמו! (גם השלישי לשני).



פתרון שאלה 16

א. טבלת מעקב אחר ביצוע האלגוריתם I

שורה	n	sum	gil	(gil > 6) או (gil < 16)	פלט
(1)	7	0			
(2)-(2.1)-(2.2)		1	15	אמת (אמת או אמת) ($15 > 6$ או $15 < 16$)	
(2)-(2.1)-(2.2)		2	10	אמת (אמת או אמת) ($10 > 6$ או $10 < 16$)	
(2)-(2.1)-(2.2)		3	19	אמת (אמת או שקר) ($19 > 6$ או $19 < 16$)	
(2)-(2.1)-(2.2)		4	18	אמת (אמת או שקר) ($18 > 6$ או $18 < 16$)	
(2)-(2.1)-(2.2)		5	7	אמת (אמת או אמת) ($7 > 6$ או $7 < 16$)	
(2)-(2.1)-(2.2)		6	16	אמת (אמת או שקר) ($16 > 6$ או $16 < 16$)	
(2)-(2.1)-(2.2)		7	8	אמת (אמת או אמת) ($8 > 6$ או $8 < 16$)	
(3)					7

אלגוריתם זה מונה ומדפיס למעשה את מספר הקלטים שהוא n. כל מספר שיקלט יהיה גדול מ-6 או קטן מ-16. לכן אלגוריתם זה לא מבצע את המטרה בשאלה.

טבלת מעקב אחר ביצוע האלגוריתם II

שורה	n	sum	gil	(gil > 6) וגם (gil < 16)	פלט
(1)	7	0			
(2)-(2.1)-(2.2)		1	15	אמת (אמת וגם אמת) (15>6 וגם 15<16)	
(2)-(2.1)-(2.2)		2	10	אמת (אמת וגם אמת) (10>6 וגם 10<16)	
(2)-(2.1)-(2.2)			19	שקר (אמת וגם שקר) (19>6 וגם 19<16)	
(2)-(2.1)-(2.2)			18	שקר (אמת וגם שקר) (18>6 וגם 18<16)	
(2)-(2.1)-(2.2)		3	7	אמת (אמת וגם אמת) (7>6 וגם 7<16)	
(2)-(2.1)-(2.2)			16	שקר (אמת וגם שקר) (16>6 וגם 16<16)	
(2)-(2.1)-(2.2)		4	8	אמת (אמת וגם אמת) (8>6 או 8<16)	
(3)					4

אלגוריתם זה **כן** מונה ומדפיס כמה תלמידים הם בין הגילים 6 עד 16 (לא כולל). גיל שיענה על התנאי הכפול ויביא לקידום המונה עונה למעשה על התנאי: **(gil > 6) וגם (gil < 16)** ללא שוויון באף קצה.

- ♥ המקרה היחיד בו ערכו של תנאי מורכב עם הקשר **או** יהיה שקר הוא כאשר כל מרכיביו הם שקר.
- ♥ המקרה היחיד בו ערכו של תנאי מורכב עם הקשר **וגם** יהיה אמת הוא כאשר כל מרכיביו הם אמת.



הקשר הלוגי המתקיים בין תנאים מכווננים הוא קשר **וגם**, יש להקפיד ולהתייחס כך לחישוב ערכי הביטוי הלוגי



ב. פיתוח ויישום של האלגוריתם

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
n	שלם	מספר נתוני הגילים המתקבלים	שלם חיובי
sum	שלם	מונה מספר התלמידים שגילם בין 6 ל-16 לא כולל	0..n
gil	שלם	קלט תורן של גיל תלמיד	שלם חיובי
max	שלם	הגיל המקסימלי המתקבל מן הקלט בטווח הגילים בין 6 ל-16, לא כולל	7..15

טבלת תת-משימות

מטרת תת-המשימה	סוג הפעולה
ביצוע לולאה עבור קלטים	לולאה
מניית הגילים העונים על התנאי	תנאי
עדכון המקסימלי מבין הנתונים העונים על התנאי	השמה בתוך תנאי
הדפסת המקסימלי אם התקבל	תנאי מחוץ ללולאה

אלגוריתם לפתרון הבעיה

- (1) אתחל את sum ל-0
- (2) אתחל את max ל-0
- (3) בצע n פעמים:

(1.3) קלוט מספר למשתנה gil

(2.3) אם $(gil > 6)$ וגם $(gil < 16)$ אזי

(1.2.3) הגדל את sum ב-1

(2.2.3) אם $gil > max$ אזי

(1.2.2.3) השם ב-max את gil

(4) הצג כפלט 'מספר התלמידים בתחום הוא: ' sum,

(5) אם $max > 0$ אזי

(1.5) הצג כפלט 'הערך הכי גדול בתחום הוא ' max,

האלגוריתם בנוי על אלגוריתם II מן השאלה. אלגוריתם זה עונה על תנאי המניה. התנאי כאן הוא תנאי מורכב עם הקשר הלוגי וגם במקום תנאי מכוון. כמו כן בתוך התנאי יש בדיקה ועדכון של המקסימלי, והדפסתו בתום תהליך הקלט – אם קיים.



פתרון נקי חייב לקחת בחשבון את המקרה בו לא התקבל כלל ערך בתחום הנדרש. במקרה כזה אין להדפיס את המקסימלי אחרת יודפס ערך ואתחול שלו – 0.



טעות לבדוק את המקסימלי לגבי כל קלט, בדיקת המקסימלי היא רק על ערכים בתחום הנדרש.



מימוש האלגוריתם

// q16t00.c

```
/* קלט: n גילים של תלמידים */
/* פלט: מספר התלמידים בין הגילים ל-16 לא כולל והמקסימליביניהם */
public static void Main()
{
    int n, sum, max, gil, i;
    n = int.Parse(Console.ReadLine()); /* קליטת מספר הגילים */
    sum = 0; /* אתחול מונה מספר התלמידים בתחום */
    max = 0; /* אתחול הגיל המקסימלי בתחום */
    for ( i = 0; i < n; i++ ) /* לולאת קלט וטיפול בכל גיל */
    {
        gil = int.Parse(Console.ReadLine()); /* קלט גיל תורן */
        if ( (gil > 6) && (gil < 16) ) /* תנאי על תחום הגילים הנדרש */
        {
            sum ++; /* קידום מונה התלמידים בתחום */
            if (gil > max) /* בדיקה ועדכון של המקסימלי בתחום */
                max = gil;
        }
    }
    Console.WriteLine ("The number of students in age between 6 to 16 is {0} \n", sum); /*
    פלט של המונה */
    if (max > 0) /* פלט מקסימלי אם קיים */
        Console.WriteLine ("The max age between 7-15 is {0} \n", max);
}
```


פתרון שאלה 17

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	שלם	מערך הנתונים בגודל 80	שלם
i	שלם	מציין בתוך מערך a	0..79
k	שלם	מספר מציין בתחום המערך	0..79
sum1	שלם	סכום k האיברים הראשונים במערך	שלם
sum2	שלם	סכום שאר האיברים במערך	שלם

טבלת תת-משימות

מטרת תת-המשימה	סוג הפעולה
קליטת 80 ערכים למערך	לולאת מונה
קליטת k כולל בדיקת תקינות המשך קלט עד לקבלת קלט תקין	לולאת תנאי
חישוב סכום k המספרים הראשונים במערך	לולאת מונה
חישוב סכום יתר המספרים במערך	לולאת מונה
השוואה בין הסכומים והצגת הוצעה בפלט	תנאי

אלגוריתם לפתרון הבעיה

- (1) עבור i מ-0 עד 79 בצע
 - (1.1) קלוט מספר למשתנה a [i]
 - (2) קלוט מספר למשתנה k
 - (3) כל עוד $(k < 0)$ או $(k > 79)$ בצע
 - (1.3) קלוט מספר למשתנה k
 - (4) אתחל את sum1 ל-0
 - (5) עבור i מ-0 עד $(k - 1)$ בצע
 - (1.5) הגדל את sum1 ב- a [i]
 - (6) אתחל את sum2 ל-0
 - (7) עבור i מ-k עד 79 בצע
 - (1.7) הגדל את sum2 ב- a [i]
 - (8) אם $sum1 > sum2$ אזי
 - (1.8) הצג כפלט: sum1

יש לבצע בדיקת תקינות על הקלט k. טווח הערכים שלו הוא בין 0..79. הקצוות בהחלט תקינים: עבור $k = 0$ יש 0 איברים בחלק הראשון ו-80 איברים בחלק השני; עבור $k = 79$ יש 80 איברים בחלק הראשון ו-0 איברים בחלק השני. לולאת הקלט ב-3 תבצע שוב ושוב כל עוד הערך יהיה קטן מן הנדרש או גדול מן הנדרש



יש לשים לב שהלולאה הראשונה מסתיימת באיבר שמספרו הסידורי $(k-1)$ והלולאה השנייה מתחילה באיבר שמספרו הסידורי k, אחרת האיבר שמציינו k ייצבר בשני הסכומים.



// q17t00.c

```

/* קלט : מערך בגודל 80 ומציין k במערך */
/* פלט : הדפסת סכום האיברים הראשונים במערך אם הוא גדול מסכום האיברים האחרונים */
public static void Main()
{
    int a[] = new int[80];
    int k, i;
    int sum1 = 0, sum2 = 0; /* אתחול סכום "האיברים הראשונים" וסכום "שאר האיברים" במערך */

    for (i = 0; i < a.Length; i++) /* קלט למערך */
        a[i] = int.Parse(Console.ReadLine());
    Console.WriteLine ("Enter k : ");
    k = int.Parse(Console.ReadLine()); /* קליטת k עד לקבלת ערך בתחום הנדרש */
    while ( (k < 0) || (k >= 80) )
    {
        Console.WriteLine ("Enter k : ");
        k = int.Parse(Console.ReadLine());
    }
    for (i = 0; i < k; i++) /* לולאה לצבירת הסכום של האיברים הראשונים במערך */
        sum1 += a [i];
    for (i = k ; i < a.Length; i++) /* לולאה לצבירת הסכום של שאר המערך */
        sum2 += a [i];
    if (sum1 > sum2) /* השוואה הסכומים ופלט נדרש */
        Console.WriteLine ("%d \n", sum1);
}

```

שים ♥ : אם $k = 0$ אזי הלולאה הראשונה צריכה להתבצע בין 1 ל-0, מאחר וזה טווח לא חוקי היא לא תתבצע כלל וערכו של sum1 יישאר 0 כפי שצריך.

פתרון שאלה 18

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
num	שלם	מספר אישי של תלמיד-קלט	1000-5000
counter	שלם	מונה את מספר התלמידים היוצאים לטיול	0..4001
price	שלם	מחיר מחושב לתלמיד לפי כמות התלמידים	100,120,150
sum	שלם	סך כל הוצאות ביה"ס עבור הטיול	שלם חיובי

טבלת תת-משימות

מטרת תת-המשימה	סוג הפעולה
קליטת המספרים האישיים של התלמידים ומניית מספר היוצאים	לולאת זקיף
בדיקת המחיר לתלמיד על פי מספר היוצאים	תנאי מקונן
חישוב הסכום הכולל	השמה
הדפסת מספר התלמידים המחיר לתלמיד ומחיר הטיול	הדפסה
בדיקת חריגה מתקציב ביה"ס והדפסה בהתאם	תנאי

אלגוריתם לפתרון הבעיה

- (1) השם 0 ב- counter
- (2) קלוט מספר אישי למשתנה num
- (3) כל עוד ($num \leq 5000$) וגם ($1000 \leq num$) בצע
 - (1.3) הגדל את counter ב-1
 - (2.3) קלוט מספר אישי למשתנה num
 - (4) אם $counter \leq 100$
 - (1.4) השם 150 ב- price
- אחרת
 - (2.4) אם $counter \leq 200$
 - (1.2.4) השם 120 ב- price
- אחרת
 - (2.2.4) השם 100 ב- price
- (5) השם $counter * price$ ב- sum
- (6) הצג כפלט 'מספר התלמידים שנרשמו לטיול: ', counter
- (7) הצג כפלט 'המחיר לתשלום לתלמיד: ', price
- (8) הצג כפלט 'עלות הטיול הכוללת: ', sum
- (9) אם $sum < 25000$ אזי
 - (1.9) הצג כפלט 'יש חריגה מן התקציב ב-: ', ($sum - 25000$)

מימוש האלגוריתם

```
// q18t00.c
void main()
{
    int counter, num, price, sum;
    counter = 0; // אתחול מונה מספר התלמידים */
    Console.WriteLine ("Enter student numbers:");
    num = int.Parse(Console.ReadLine()); // קליטת מספר אישי של תלמיד */
    while ((1000<= num)&&(num<=5000)) // כל עוד המספר האישי הנקלט בתחום */
    {
        counter ++; // קדם את מונה התלמידים */
        num = int.Parse(Console.ReadLine()); // קליטת מספר אישי של תלמיד */
    }
    if (counter <= 100) // אם מספר התלמידים קטן או שווה ל-100 */
        price = 150; // המחיר לתלמיד הוא 150 ש"ח */
    else
        if (counter <= 200) // אם מספר התלמידים גדול מ-100 אך קטן או שווה ל-200 */
            price = 120; // המחיר לתלמיד הוא 120 ש"ח */
        else
            price = 100; // אם מספר התלמידים הוא מעל 200 המחיר לתלמיד הוא 100 ש"ח */
    sum = counter * price; // חישוב עלות הטיול הכוללת */
    Console.WriteLine ("number of students: {0}\n", counter); // הדפסת מספר התלמידים */
    Console.WriteLine ("price for student : {0} \n", price); // הדפסת המחיר לתלמיד */
    Console.WriteLine ("total sum is : {0} \n", sum); // הדפסת הסכום הכולל */
    if (sum > 25000) // אם יש חריגה מן התקציב הדפס אותה */
        Console.WriteLine ("There is an exception of : %d \n", (sum - 25000));
}
}
```

בשאלה מצויין במפורש תנאי סיום הקלט: "הקליטה תסתיים כאשר מספר התלמיד הנקלט הוא מחוץ לתחום", כלומר לא ערך מסוים המשמש כזקיף. לכן תנאי הלולאה צריך להיות: "כל עוד נקלט מספר בתחום המשך לבצע". הקשר הלוגי בין התנאים חייב להיות 'וגם' כדי לתאר מספר בתחום.



מבנה התנאי מתבסס על שימוש בתנאי מ קונן – כלומר שימוש ב else. במבנה זה אין צורך להשתמש בתנאי מורכב. אם משתמשים בתנאים פשוטים באופן לא מקונן תתקבל טעות. למשל אם רושמים את התנאים באופן הבא:



```
if ( counter <= 100 )
    price = 150;
if ( counter <= 200 )
    price = 120;
```

עבור ערך הקטן מ-100, יתקיים התנאי הראשון וגם התנאי השני אחריו, והערך הסופי של price יהיה 120 במקום 150.

אם כותבים תנאים לא מקוננים הם חייבים להיות מורכבים למשל:

```
if (counter <= 100)
    price = 150;
if ( ( counter > 100 ) && ( counter <= 200 ) )
    price = 120;
```

מספר פתרונות



פתרון שאלה 19

א. פיתוח ויישום של פונקציה שמקבלת מספר תלת ספרתי ומחזירה את המשקל שלו

טבלת משתנים לפונקציה `numberWeight`

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
num	שלם	פרמטר, מספר תלת-ספרתי	0..999
dig1	תחום	ספרת האחדות של num	0..9
dig2	תחום	ספרת העשרות של num	0..9
dig3	תחום	ספרת המאות של num	0..9

טבלת תת-משימות לפונקציה `numberWeight`

מטרת תת-המשימה	סוג הפעולה
פירוק ספרת האחדות	חישוב
פירוק ספרת העשרות	חישוב
פירוק ספרת המאות	חישוב
הרכבת המשקל של הפרמטר	חישוב

אלגוריתם לפתרון הפונקציה

(1) השם את ספרת האחדות ב-dig1

(2) השם את ספרת העשרות ב-dig2

(3) השם את ספרת המאות ב-dig3

(4) החזר את $dig3 * dig2 + dig2 * dig1$

מימוש הפונקציה

```
public static int numberWeight(int num)
    /* טענת כניסה: הפונקציה מקבל מספר תלת-ספרתי num */
    /* טענת יציאה: הפונקציה מחזירה את המשקל של num */
{
    int dig1, dig2, dig3;
    dig1 = num % 10; /* פירוק ספרת האחדות */
    dig2 = (num / 10) % 10; /* פירוק ספרת העשרות */
    dig3 = num / 100; /* פירוק ספרת המאות */
    return ((dig3 * dig2) + (dig2 * dig1)); /* החזרת משקל המספר */
}
```

ב. פיתוח ויישום של אלגוריתם שמקבל מספרים תלת ספרתי ומחשב את המשקל שלהם

טבלת משתנים לתכנית הראשית

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
number	שלם	נתון קלט תורן- מספר תלת-ספרתי	0..999
sum	שלם	סכום המשקלים של משתני הקלט	שלם

טבלת תת-משימות

סוג הפעולה	מטרת תת-המשימה
לולאת תנאי	קליטת מספרים תלת-ספרתיים עד שסכום משקליהם עולה על 100
פונקציה	חישוב "משקל" של מספר

אלגוריתם לפתרון הבעיה

- (1) השם 0 ב-sum
- (2) כל עוד $sum \leq 100$ בצע
 - (1.2) קלוט מספר תלת-ספרתי למשתנה number
 - (2.2) הגדל את sum ב-"משקל" המספר number
- (3) הצג כפלט את סכום המשקלים שהתקבל ב-sum

מימוש האלגוריתם

```

/* קלט : מספרים תלת ספרתיים */
/* פלט : הדפסת סכום המשקלים של מספרי הקלט עד שהוא עולה על 100 */
/* הגדרה של משקל: (מכפלת ספרה ראשונה בספרה שנייה) + (מכפלת ספרה שנייה בספרה שלישית) */

```

// q19t00.c

```

public static int numberWeight (int num) /* הפונקציה מסעיף א */
{
    int dig1, dig2, dig3;
    dig1 = num % 10;
    dig2 = (num / 10) % 10;
    dig3 = num / 100;
    return ((dig3 * dig2) + (dig2 * dig1));
}

public static void Main()
{
    int number, sum = 0; /* אתחל הסכום ל 0 */
    while (sum <= 100) /* כל עוד הסכום קטן או שווה ל 100 */
    {
        Console.WriteLine ("Enter number : \n");
        number = int.Parse(Console.ReadLine()); /* קלוט נתון תורן */
        sum += numberWeight (number); /* הוסף את משקל הנתון התורן לסכום המצטבר */
    }
    Console.WriteLine ("The weights sum is: {0}\n", sum); /* הדפס את הסכום שהתקבל */
}

```

הלולאה הראשית באלגוריתם היא לולאת תנאי. למרות שבכל ביצוע של הלולאה יש קלט של נתון חדש אין זו לולאת זקיף. סיום הלולאה אינו מותנה בקבלת קלט חריג (זקיף) אלא בקיום תנאי על הסכום המצטבר של "משקלי" מספרי הקלט. על פי נתוני השאלה "הקליטה תסתיים כאשר סכום המשקלים יהיה גדול מ-100" – כלומר בפעם הראשונה שסכום המשקלים גדול מן הערך 100 – תיפסק קליטת הנתונים. מכאן שאם הסכום האחרון עדיין קטן או שווה ל-100, יש לקלוט מספר חדש ולהוסיף את משקלו לסכום





א. פיתוח ויישום של פונקציה שמקבלת כקלט מערך בן 10 איברים...

טבלת משתנים לפונקציה up1 (גרסת פתרון 1)

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	מערך	פרמטר, מערך בן 10 איברים	שלמים
i	שלם	מציין המתקדם על המערך	0..9
ok	שלם	מסמן האם המערך עולה או לא דגל. כל זמן שחלק המערך הנבדק הוא עולה יהיה ערכו, כאשר מתגלה כי המערך אינו עולה יהיה ערכו 0.	1, 0

אלגוריתם לפתרון הבעיה

מערך עולה (a)

/* טענת כניסה: הפעולה מקבלת מערך חד-מימדי בגודל 10 של מספרים שלמים */

/* טענת יציאה: הפעולה מחזירה 1 אם המערך ממוין בסדר עולה ו 0 אחרת */

(1) השם 1 ב- ok

(2) השם 1 ב- i

(3) כל עוד (i < 10) וגם ok בצע

(1.3) אם a[i] >= a[i+1] אזי

(1.1.3) השם 0 ב- ok

אחרת

(2.1.3) הגדל את i ב-1

(4) החזר את ok

מימוש האלגוריתם

public static int up1(arr a) /* טענת כניסה: הפונקציה מקבלת מערך חדמימדי a בגודל 10

/* טענת יציאה: הפונקציה מחזירה אמת אם המערך עולה ו שקר אחרת */

```

{
  int i, ok;
  ok = 1; /* אתחול הדגל ל'אמת' */
  i = 0; /* אתחול i לסריקת המערך */
  while ( ( i < 9) && (ok ==1) ) /* כל עוד המערך תקין וגם יש עוד זוג לבדוק */
  {
    if ( a [i] >= a [i+1] ) /* אם הזוג הנוכחי אינו מסודר -עדכן הדגל ל'שקר' */
      ok = 0;
    else /* אחרת התקדם לזוג הבא */
      i ++;
  }
  return ok; /* החזרת ערך המסקנה ok */
}
    
```



הפתרון משתמש במשתנה סימון ok מקומי לפונקציה, המעיד על תקינות המערך עד למיקום i בהתאמה. כאשר מתגלה אי-תקינות הוא מקבל ערך 'שקר' - 0, הלולאה מסתיימת וערכו מוחזר על ידי הפונקציה.

מספר פתרונות



טבלת משתנים לפונקציה up2 (גרסת פתרון 2)

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	מערך	פרמטר, מערך בן 10 איברים	שלמים
i	שלם	מציין המתקדם על המערך	0..9

אלגוריתם לפתרון הבעיה

- (1) השם 0 ב- i
- (2) כל עוד $(i < 10)$ וגם $(a[i] < a[i+1])$ בצע
 - (1.2) הגדל את i ב-1
- (3) אם $i = 10$ אזי
 - (1.3) החזר 1 אחרת
 - (2.3) החזר 0

מימוש האלגוריתם

```

public static
int up2(arr a) /* טענת כניסה: הפונקציה מקבלת מערך חד-מימדי a בגודל 10 */
/* טענת יציאה: הפונקציה מחזירה 'אמת' אם המערך עולה ו'שקר' אחרת */
{
    int i;
    i = 0; /* אתחול i לסריקת המערך */
    while ( (a[i] < a[i+1]) && (i < 9)) /* כל עוד המערך עולה ואין חריגה מגבולות המערך */
        i++; /* התקדם במערך */
    if ( i == 9) /* אם הגעת לסוף הסריקה אזי החזר 'אמת' */
        return 1;
    else /* אחרת החזר 'שקר' */
        return 0;
}
    
```

פתרון זה אינו משתמש במשתנה נוסף. בדיקת הרצף העולה נעשית בתוך תנאי הלולאה. כאשר הלולאה מסתיימת לא ידוע האם היא הסתיימה כי נבדקו כל הערכים או כי התגלתה אי תקינות. מאחר והזוג האחרון הנבדק הוא עבור $i = 8$, כלומר נבדקים האיברים במקומות 9, 8, ואז ערכו של i מקודם ל-9. כלומר, ערכו של i מעיד על תקינות כל המערך. אם ערכו הגיע ל-10, אזי המערך תקין ומוחזר 'אמת'. אחרת המערך אינו תקין ומוחזר 'שקר'.



ב. פיתוח ויישום אלגוריתם שקולט מספרים למערך בן 120 איברים.

פתרון גרסה 1:

פתרון זה מבוסס על פיתוח הפונקציה שנדרשה בסעיף א כלשונה. כלומר הפונקציה מקבלת מערך בגודל 10 ובודקת האם הוא עולה. מאחר והתכנית הראשית משתמשת במערך גזול יותר, ויש לבצע את העיבוד על חלקים שלו. לכן יש שימוש במערך עזר a בגודל 10 שמטרתו להעביר את קבוצת הנתונים הנדרשת לבדיקה לפונקציה



טבלת תת-משימות

מטרת תת-המשימה	סוג הפעולה
בדיקה האם במערך בגודל 10 מתקיים סדר עולה	פונקציה
קליטת מערך בגודל 12	לולאה
12 פעמים (120/10) יש לבצע: - העתקת 10 נתונים מן המערך הגדול לתוך מערך ב-10 איברים לצורך בדיקת הסדר העולה - בדיקת הסדר העולה ב-10 הערכים התורנים - קידום מונה מספר הסדרות העולות	לולאה ראשית: - לולאה - הפעלת הפונקציה - תנאי וקידום
הדפסת מונה מספר הסדרות העולות	פלט

טבלת משתנים לאלגוריתם הראשי

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
bigA	מערך	המערך הנבדק בגודל 120	שלמים
a	מערך	מערך בן 10 איברים לצורך העברת נתונים לפונקציה	שלמים
i	שלם	מציין את מספר סדרת הנתונים הנבדקת	1..12
j	שלם	מציין במערך a	0..9
counter	שלם	מונה את מספר הסדרות העולות במערך bigA	1..12

אלגוריתם לפתרון הבעיה

- (1) קלוט 120 נתונים למערך bigA
- (2) אתחל את counter ל-0
- (3) עבור i מ-0 עד 11 בצע
 - (1.3) העתק את קבוצת הנתונים ה-i ממערך bigA למערך a
 - (2.3) אם קבוצת הנתונים במערך a היא עולה
 - (1.2.3) קדם את counter ב-1
- (4) הצג כפלט את counter

מימוש האלגוריתם

```
// q20at00.c
/* קלט : מערך בגודל 120 שלמים
 * פלט : מספר תתי המערכים בגודל 10 המקיימים סדר עולה
 * הפונקציה מסעיף א- על פי הגרסה הנבחרת */

public static int up1(int[] a)
{
    int i, ok;
    ok = 1;
    i = 0;
    while ( ( i < 9 ) && ( ok == 1 ) )
    {
        if ( a[i] >= a [i+1] )
            ok = 0;
        else
            i ++;
    }
    return ok;
}

public static int up2(int[] a)
{
    inti;

```

```

i = 0;
while ( ( a [i] < a [i+1]) && (i < 9) )
    i ++;
if ( i == 9 )
    return 1;
else
    return 0;
}
public static void readArr (int[] a)
/* טענת כניסה ויציאה: הפונקציה קולטת ערכים למערך a המועבר כפרמטר */
{
int i;
Console.WriteLine ("Enter big array elements ");
for ( i = 0; i < 120; i ++ )
    a[i] = int.Parse(Console.ReadLine());
}
public static void Main ()
{
inta = new int[10];
int[] bigA = new int[120];
int counter, i, j;

readArr (bigA);
counter = 0;
for ( i = 0; i < 12; i ++ )
{
for ( j = 0; j < 10; j ++ )
    a [j] = bigA [i*10 + j] ;
if (up2 (a) == 1)
    counter ++;
}
Console.WriteLine ("Number of getting up arrays is : {0} ", counter );/*
}

```

פתרון גרסה 2 :

פתרון זה מבוסס על פיתוח הפונקציה שנדרשה בסעיף א באופן המותאם לביצוע המשימה הכוללת של האלגוריתם. כאן הפונקציה תפעל על מערך בגודל 120 איברים והיא תקבל פרמטר נוסף שמטרתו לציין את תחילת הקטע הנבדק. גם כאן הבדיקה תהיה בכל פעם האם קטע של 10 נתונים הוא עולה. פתרון זה משחרר מן הצורך להשתמש במערך קטן בן 10 איברים לצורך התאמה לפונקציה, ומן הצורך להעתיק בכל פעם 10 נתונים.



טבלת תת-משימות

סוג הפעולה	מטרת תת-המשימה
פונקציה	בדיקה האם בקטע מערך בגודל 10 מתקיים סדר עולה
לולאה	קליטת מערך בגודל 12
לולאה:	12 פעמים (12/10) יש לבצע:
- הפעלת הפונקציה	- בדיקת הסדר העולה בתת המערך התורן בגודל 10.
- תנאי וקידום	- קידום מונה מספר הסדרות העולות
פלט	הדפסת מונה מספר הסדרות העולות

טבלת משתנים לאלגוריתם הראשי

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים

שלמים	המערך הנבדק בגודל 120	מערך	biga
0..11	מציין את מספר סדרת הנתונים הנבדקת	שלם	i
1..12	מונה את מספר הסדחת העולות במערך biga	שלם	counter

אלגוריתם לפתרון הבעיה

(1) קלוט 120 נתונים למערך biga

(2) אתחל את counter ל-0

(3) עבור i מ-1 עד 12 בצע

(1.3) אם קבוצת הנתונים במערך a היא עולה

(1.1.3) קדם את counter ב-1

(4) הצג כפלט את counter

```
// q20b00;
/* קלט : מערך בגודל 120 שלמים */
/* פלט : מספר תתי המערכים בגודל 10 המקיימים סדר עולה */
typedef int bigArr [120];

public static int up (int[] a, int p)
/* טענת כניסה: הפונקציה מקבלת מערך a ומציין במערך p מסודרים בסדר עולה */
/* טענת יציאה: הפונקציה מחזירה 'אמת' אם 10 הנתונים במערך החל ממקום p מסודרים בסדר עולה */
{
    int i;
    i = p;
    while ( (a [i]< a [i+1]) && (i < p+9) ) /* התחל את בדיקת הקטע ממיקום p */
        i ++; /* כל עוד הערכים עולים ולא נבדקו 10 נתונים */
    if ( i == p+9 ) /* התקדם בבדיקה */
        return 1; /* אם הבדיקה הסתיימה בהצלחה החזר 'אמת' */
    else /* אחרת החזר 'שקר' */
        return 0;
}

public static void readArr (bigArr a)
/* טענת כניסה ויציאה: הפונקציה קולטת ערכים למערך a המועבר כפרמטר */
{
    int i;
    Console.WriteLine("Enter big array elements \n");
    for ( i = 0; i < 120; i++)
        a[i] = int.Parse(Console.ReadLine());
}

public static void Main()
{
    int[] bigA = new int[120];
    int counter, i;
    readArr (bigA); /* קליטת מערך הנתונים */
    counter = 0; /* אתחול מונה מספר הסדרות ל-0 */
    for ( i = 0; i < 12; i++) /* טיפול בכל אחת מ-12 סדרות הנתונים */
        if ( up(bigA, i*10) == 1 ) /* בדיקה האם הקבוצה התורנית מקיימת את התנאי */
            counter ++; /* אם כן, קידום המונה */
    Console.WriteLine("\nNumber of getting up arrays {0} ", counter ); /* הדפסת המונה */
}

```