

# מבחן 8

## מבחן בגרות היצוני - שנת 2001

### פרטים כלליים

מועד הבחינה : קיץ תשס"א, 2001

מספר השאלון : 899222

משך הבחינה : שלוש שעות

חומר עזר בשימוש : כל חומר עזר (חוץ ממחשב הניתן לתכנות)

המלצות : קרא המלצות לפני הבחינה ובדיקות אחרונות לפני מסירה (עמודים 8-10)

שים ♥ : בבחינה זו נציג פתרונות לשאלות 11-20 שמתאימים לתכנית הלימודים החדשה

### מבנה השאלון ומפתח הערכה

פרק ראשון	יש לענות על השאלות 1-5 לכל שאלה – 10 נקודות	סה"כ 50 נקודות
פרק שני	יש לענות על <u>שתיים</u> מהשאלות 6-8 לכל שאלה – 15 נקודות	סה"כ 30 נקודות
פרק שלישי	יש לענות על <u>אחת</u> מהשאלות 9-10 לשאלה – 20 נקודות	סה"כ 20 נקודות

### תוכן עניינים של פתרון המבחן

- שאלה 11 : לולאת מונה – נתון אלגוריתם, מה הפלט ? ..... 149
- שאלה 12 : הוראת תנאי – נתונים קטעי תכנית, מה הקלט לפלט נתון ? ..... 149
- שאלה 13 : מערך דו-מימדי, לולאות – מה הפלט ? ..... 150
- שאלה 14 : לולאת מונה, מחרוזות – כתוב תכנית ..... 151
- שאלה 15 : מערך חד-מימדי – כתוב קטעי תכנית ..... 151
- שאלה 16 : פונקציות, הרכבת מספר – מה מבצעת פונקציה ?, שינוי בפונקציה ..... 152
- שאלה 17 : לולאת תנאי (זקיף) – כתוב תכנית ..... 154
- שאלה 18 : מערך חד-מימדי, מקסימלי – פתח אלגוריתם, כתוב תכנית ..... 155
- שאלה 19 : פונקציות, מחרוזות – תת-משימות, תת-תכניות, תכנית ראשית ..... 157
- שאלה 20 : פונקציות, חישוב – תת-משימות, תת-תכניות, תכנית ראשית ..... 161

# פתרון מבחן בגרות תשס"א 2001

## פתרון 8

### פתרון פרק ראשון - 50 נקודות

#### פתרון שאלה 11

א. טבלת מעקב עבור הקלט  $n = 4$  ו-  $number = 7$  :

מספר ביצוע של הלולאה	tot	n	number	פלט
ערכים התחלתיים	0	4	7	
לולאה	1	$(0+7=)$ 7		
	2	$(7+7=)$ 14		
	3	$(14+7=)$ 21		
	4	$(21+7=)$ 28		
				28

ב. עבור הקלט 3 ל-  $n$  והקלט 10 ל-  $number$  יהיה הפלט המספר 30.

מספר ביצוע של הלולאה	tot	n	number	פלט
ערכים התחלתיים	0	3	10	
לולאה	1	$0+10=10$		
	2	$10+10=20$		
	3	$20+10=30$		
				30

האלגוריתם מחשב ומציג כפלט את הערך של המכפלה  $number * n$ . מאחר ואל הצובר tot מוסיפים  $n$  פעמים את  $number$ . לכן כל שני מספרים שמכפלתם 30 מתאימים כקלטים לשני המשתנים.



#### פתרון שאלה 12

(I) הפלט BFB מתקבל עבור הערך 'B' למשתנה AV2 והערך 'F' למשתנה TAV1. שורת האלגוריתם המוצגת כפלט היא (3.1).

מאחר ובפלט מוצגים שני תווים שונים והראשון זהה לאחרון – קיימות שתי שורות באלגוריתם המתאימות לפלט והן : (3.1), (3.2.2). על פיהן ברור ש-  $TAV1='F'$  ו-  $TAV2='B'$ . אם כך נעקוב אחר התנאים באלגוריתם: התנאי (3) – מתקיים לכן מתבצעת שורת הפלט (3.1).



TAV1	TAV2	התנאי TAV1 > TAV2	התנאי TAV2 > 'D'	פלט
'F'	'B'	אמת 'F' > 'B'		
			לא נבדק	
				BFB

(II) הפלט LLL מתקבל עבור הערך 'L' לשני המשתנים TAV1 ו-TAV2. שורת האלגוריתם המוצגת כפלט היא (3.2.1).

מאחר ובפלט מוצגים שלושה תווים זהים – ברור ששני התווים זהים ושווים ל 'L' כי בכל אחת משורות הפלט מוצגים שני המשתנים. אם כך נעקוב אחר התנאים באלגוריתם: התנאי (3) – אינו מתקיים לכן עוברים לאחרת; התנאי (3.2) – מתקיים לכן מתבצעת שורת הפלט (3.2.1).



TAV1	TAV2	התנאי TAV1 > TAV2	התנאי TAV2 > 'D'	פלט
'L'	'L'	שקר 'L' > 'L'		
			אמת 'L' > 'D'	
				LLL



### פתרון שאלה 13

א. טבלת מעקב

המעקב מתבצע על מערך דו-מימדי, קיימים שני מציינים, אך אין לולאות מקוננות ולא מנגנון המדמה לולאות מקוננות על ידי לולאה אחת, לכן אין לצפות לפלט של כל המערך. למעשה בשורת הפלט (2.2), ערכו של i תמיד זהה לערכו של j. i מקודם על ידי הלולאה ו- j מקודם באופן "ידני". ומאחר והלולאה מתבצעת 3 פעמים יודפסו הערכים שהמציינים שלהם הם: (1,1), (2,2), (3,3)

i	j	פלט j a [i, j]
	1	
1	2	1 3
2	3	2 6
3	4	3 8

ערכים התחלתיים

לולאה

ב.

שינוי שורת האלגוריתם מביא לכך שבשורת הפלט (2.2), ערכו של j תמיד זהה ל-1. מאחר ו- i מתקדם מ-1 עד 3, אזי יש התקדמות על פני השורות אך העמודה נשארת קבועה – העמודה הראשונה. יודפסו הערכים שהמציינים שלהם הם: (1,1), (2,1), (3,1).

i	j	פלט j a [i, j]
	1	
1	1	3
2	1	7
3	1	1

ערכים התחלתיים

לולאה

הסבר הפתרון

## מספר פתרונות



## פתרון שאלה 14

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
line	מחרוזת בגודל 15 תווים	מחרוזת קלט תורנית לבדיקה	תווים
i	שלם	משתנה לולאה לקליטת 200 מחרוזות	1..200

// q14t01.c

```
public static void Main(string[] args)
```

```
{
    char line [15];
    int i;
    for (i = 0; i < 200; i++)
    {
        Console.Write ("enter line: ");
        line = Console.ReadLine();
        if ( (line [0]=='w') && (line [1]=='w') && (line [2]=='w') )
            Console.WriteLine (line);
    }
}
```

```
/* קלט : 200 מחרוזות באורך 15 תווים */
/* פלט : המחרוזות המתחילות ב 'www' */
```



מחרוזת היא מערך של תווים ולכן הפניה אל תווים בודדים במחרוזת היא פנייה על פי המצייך של התו. התו הראשון מיקומו 0. הוראת התנאי המורכבת בודקת אם כל אחד מ-3 התווים הראשונים הוא התו 'w'.

## פתרון שאלה 15

א. טבלת משתנים

תכנית

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	מערך חד-מימדי	המערך הנתון	שלמים
i	שלם	משתנה לולאה לסריקת המערך	10..998

```
public static void main(string[] args)
{
    int a [999];
    int i;
    for ( i = 0; i < 999; i++)
        a [i] =
int.Parse(Console.ReadLine());
    for ( i = 10; i < 999; i += 2)
    {
        Console.Write (a [i]+' ');
    }
}
```



ניתן להתקדם על המציינים הזוגיים במערך על ידי קידום מונה הלולאה ב-2 בכל פעם. בדרך זו אין צורך לבדוק זוגיות.

פתרון נוסף – בדיקת זוגיות המציין:

תכנית

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	מערך חד-מימדי	המערך הנתון	שלמים

בדיקת הזוגיות היא על המציין של התא, המשתנה i. הפלט הוא של ערך התא העונה על התנאי, המשתנה a [i].

שים



תכנית

ב. טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
b	מערך חד-מימדי	המערך הנתון	שלמים
i	שלם	משתנה לולאה לסריקת המערך	1..999

בדיקת הזוגיות היא על הערך של התא, המשתנה b [i]. הפלט הוא של הערך של התא העונה על התנאי, גם הוא המשתנה b [i].

שים



כאשר רשום בשאלה שנתון מערך אין צורך להגדיר אותו ואין צורך לקלוט נתונים. הכוונה לכתוב קטע של תכנית.



פתרון שאלה 16

טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
n	שלם	פרמטר, מספר הספרות במספר החדש num	0..4
k	שלם	פרמטר, סיפורה אחת שחוזרת במספר החדש num	1..9
i	שלם	מונה עבור מספר הספרות במספר החדש num	0..n

שם	המספר החדש שנבנה	שם	num
----	------------------	----	-----

א. /\* טענת יציאה : הפונקציה מחזירה מספר ובו n+1 ספרות וכל ספרה בו היא k \*/  
 טבלת מעקב אחר ביצוע הפונקציה הנתונה עבור ערכי הפרמטרים : n = 3 , k = 2

ערך מוחזר	i	num	i <= n	פרמטר k	פרמטר n	ערכים התחלתיים
	0	0		2	3	
	1	2 (0*10+2=)	אמת (0<=3)			לולאה
	2	22 (2*10+2=)	אמת (1<=3)			
	3	222 (22*10+2=)	אמת (2<=3)			
	4	2222 (222*10+2=)	אמת (3<=3)			
			שקר (4<=3)			
2222						

ב. קיימות שתי גרסאות לפתרון תוך שינוי קל מאד של הפונקציה המקורית :

גרסת פתרון 2:	גרסת פתרון 1:
<p>כדי שהלולאה תבצע n פעמים ניתן לאתחל את i ב-1 במקום ב-0. כך ערכי i עבורם תבצע הלולאה הם : 1..n, כלומר n פעמים.</p> <pre>int makeNumber (int n, int k) /* טענת כניסה : הפונקציה מקבלת כקלט שני n /* מספרים חד-ספרתיים n, k, n &lt;= 4, k &lt;= 9 /* טענת יציאה : הפונקציה מחזירה מספר ובו n /* n ספרות שכולן זהות לספרה k */ {   int num = 0;   <b>int i = 1;</b>   while (i &lt;= n) do   {     num = num * 10 + k;     i++;   }   return num; }</pre>	<p>כדי שהלולאה תבצע n פעמים ניתן להוריד את הסימן = מתנאי הלולאה. כך ערכי i עבורם תבצע הלולאה הם : 0..(n-1), כלומר n פעמים.</p> <pre>int makeNumber (int n, int k) /* טענת כניסה : הפונקציה מקבלת כקלט שני n /* מספרים חד-ספרתיים n, k, n &lt;= 4, k &lt;= 9 /* טענת יציאה : הפונקציה מחזירה מספר ובו n /* n ספרות שכולן זהות לספרה k */ {   int num = 0;   int i = 0;   <b>while (i &lt; n) do</b>   {     num = num * 10 + k;     i++;   }   return num; }</pre>

```
גרסת פתרון 1 : השמת ערך הפונקציה למשתנה.
new = makeNumber( (mis / 10), (mis % 10) );

גרסת פתרון 2 : הדפסת ערך הפונקציה.
Console.WriteLine ( makeNumber( (mis / 10), (mis % 10) ) );
```

**הפרמטר n**, מייצג את אורך המספר והוא המופיע הראשון משמאל, לכן יש להעביר אליו את ספרת העשרות של mis. **הפרמטר k**, מייצג את הסיפרה החוזרת על עצמה במספר והוא המופיע שני ברשימת הפרמטרים, לכן יש להעביר אליו את ספרת האחדות של mis.

**דוגמה:** אם mis = 49, אזי ערכי הפרמטרים יהיו : k = (mis % 10 = 9), n = (mis / 10) = 4.



כאשר רושמים זימון לפונקציה יש לשלב אותו בהוראה תקינה. פונקציה מחזירה ערך ובמסגרת הוראה יש לתת לערך הקשר נכון. כלומר, להשים אותו למשתנה, לשותף אותו בביטוי חשבוני תקין שמושם למשתנה, או להדפיס אותו.



## פתרון שאלה 17

### טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
name	מחרוזת עד 20 תווים	קליטת שם הבוגר	תווים
code	שלם	קליטת קוד: 1- לבד; 2- עם בן/בת זוג	1..2
children	שלם	קליטת מספר ילדים	שלם חיובי או 0
sumA	שלם	צובר את סך כל המבוגרים	שלם חיובי או 0
sumC	שלם	צובר את סך כל הילדים	שלם חיובי או 0
guides	שלם	מספר המדריכים הנדרשים (1 לקבוצה של 20 ילדים)	שלם חיובי או 0

// q17t01.c

```
/* קלט : שלשות הכוללות : שם, קוד בן זוג, מספר ילדים (עד ל finish) */
/* פלט : מספר מבוגרים ומספר מדריכים לפי 20 ילדים בקבוצה */
```

```
public static void main (string[] args)
```

```
{
    char name [20];
    int code, children, sumA, sumC, guides;
    sumA = 0; /* אתחול צובר מבוגרים */
    sumC = 0; /* אתחול צובר ילדים */
    Console.WriteLine ("enter name: ");
    name = Console.ReadLine(); /* קליטת שם הבוגר */
    while (!name.EqualsTo("FINISH")) /* כל עוד יש לקלוט פרטי בוגר */
    {
        Console.WriteLine ("enter code (1/2) and number of children:");
        code = int.Parse(Console.ReadLine()); /* קליטת קוד לבוגרים ומספר ילדים */
        children = int.Parse(Console.ReadLine()); /* קליטת קוד לבוגרים ומספר ילדים */
        sumA += code; /* צבירת מספר הבוגרים */
        sumC += children; /* צבירת מספר הילדים */
        Console.WriteLine ("enter name: ");
        name = Console.ReadLine(); /* קליטת שם הבוגר הבא */
    }
    guides = sumC / 20; /* חישוב מספר הקבוצות המלאות */
    if ((guides * 20) != children) /* עדכון מספר המדריכים אם יש צורך */
        guides ++;
    Console.WriteLine ("number of adults: "+sumA); /* פלט מספר המבוגרים */
    Console.WriteLine ("number of needed guides: "+guides); /* מספר המדריכים הנדרשים */
}
```



- הלולאה הראשית בפתרון היא לולאת זקיף מאחר ולא ידוע מראש מספר הבוגרים והקלט מתבצע עד לקליטת השם 'finish'.
- צבירת מספר המבוגרים – מאחר וכל בוגר רושם 1 אם הוא מגיע לבד ו- 2 אם הוא מגיע עם בת זוג, צבירת הנתון הזה נותנת בדיוק את מספר המבוגרים.

- חישוב מספר המדריכים – כאשר נצבר מספר הילדים יש לחלק את הסה"כ ל- 20 כדי לקבל את מספר המדריכים. יש לזכור שאין בטחון שמספר הילדים מתחלק ב- 20, ולכל מספר נוסף של ילדים יש צורך בעוד מדריך. לכן נוסף התנאי:  

```
if ( (guides * 20) != children )
    guides ++;
```
- שמשמעו: אם (מספר המדריכים שחושב על פי חלוקה ל- 20) \* 20 לא זהה למספר הילדים הכולל, סימן שיש "עודף" ילדים – ולכן צריך מדריך נוסף.
- גרסה נוספת לתנאי:  

```
if ( (guides % 20) != 0 )
    guides ++;
```

לולאת הזקיף מתבצעת על פי השם בלבד, לכן מתאים יותר לקלוט את יתר שני הנתונים בגוף הלולאה. אם השם תקין יקלטו שני הנתונים הנדרשים, אם השם מציין סיום אין טעם לקלוט את שני הנתונים.



### מספר פתרונות



פתרון חלופי לצבירת הקוד 2..1 לסך כל המבוגרים, הוא על ידי התנאי הפשוט:  

```
if (code == 1)
    suma ++;
else
    suma += 2;
```

## פתרון שאלה 18

### טבלת משתנים

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
a	מערך חד-מימדי	המערך הנתון	ממשיים
i	שלם	משתנה לולאה לסריקת המערך	0..70
pos	שלם	מיקום תורן של האיבר הראשון בזוג שסכומו המקסימלי	0..69
maxSum	ממשי	הסכום המקסימלי של זוג שהתקבל עד כה	שלם

```
// q18t01.c
/* קלט : מערך מספרים בגודל 71
/* פלט : סכום מקסימלי של זוג איברים סמוכים ומיקומיהם */
```

```
public static void Main(string[] args)
{
    float a [71];
    int i, pos;
    float maxSum;

    Console.WriteLine ("Array values:");
    for (i = 0; i < 71; i++) /* לולאה לקליטת ערכי המערך */
        a [i] = float.Parse(Console.ReadLine());
    maxSum = a [0] + a [1]; /* אתחול סכום הזוג המקסימלי התורן להיות סכום הזוג הראשון */
    pos = 0; /* אתחול של מיקום הזוג המקסימלי הראשון */
    for (i = 1; i < 70; i++) /* לולאה לסריקת הזוגות i - הראשון בזוג */
    {
```



```

if (a [i] + a [i+1] > maxSum) /* אם נמצא זוג שסכומו גבוה מן המקסימלי */
{
    maxSum = a [i] + a [i+1]; /* עדכן המקסימלי התורן */
    pos = i; /* עדכן מיקום הזוג החדש */
}
}
Console.WriteLine ("The max pear sum value is: "+maxSum); /* הדפסת סכום הזוג המקסימלי */
Console.WriteLine ("The pear places are: "+pos+", "+(pos+1));
/* הדפסת מיקומי הזוג שסכומו המקסימלי */
}

```



יש למצוא את המקסימלי בין הסכומים של זוגות איברים סמוכים במערך. כלומר עבור כל איבר, הזוג שלו נמצא במיקום עוקב לו. לכן הסריקה שצריכה להתבצע היא על ידי לולאה אחת. אם הזוגות לא היו סמוכים אזי היינו צריכים לולאה כפולה שתאפשר לייצר את כל זוגות המציינים האפשריים.

♥ חובה לאתחל את סכום הזוג המקסימלי הראשון לפני הלולאה, אחרת ההשוואה בתוך הלולאה שגוייה.

שים



♥ בכל פעם שיש עדכון של ערך הסכום המקסימלי יש לעדכן גם את מיקום הראשון בזוג. לכן יש עדכון של pos גם לפני תחילת הלולאה – בשלב האתחול וגם כאשר יש עדכון של המקסימלי בתוך התנאי. העדכון של pos חייב להיות בתוך התנאי ולא מחוצה לו – אחרת יתעדכן בכל ביצוע של הלולאה.

♥ המצייין i בלולאת הבדיקה: מתחיל ב- 2 כי זה המיקום של האיבר הראשון בזוג הבא שצריך להבדק. הזוג הראשון כבר אותחל לפני הלולאה אין כל טעם לבדוק אותו. i מסתיים ב- 69 מאחר ושם מתחיל הזוג האחרון לבדיקה. אם i יגיע לערך 70 אזי בגוף הלולאה תהיה פניה למיקום 71 – חריגה מן המערך.

אין לבדוק זוגות איברים ללא חפיפה. השאלה היא על כל הזוגות של איברים סמוכים. כלומר אם הזוג הראשון הוא במציינים  $\langle 0,1 \rangle$  אזי הזוג הבא נמצא במציינים  $\langle 1,2 \rangle$  ולא ב-  $\langle 2,3 \rangle$ . אין לדלג על הזוג  $\langle 3,4 \rangle$ .



## מספר פתרונות



**פתרון נוסף:** ניתן להשתמש בשני משתנים תורנים כמציינים של שני האיברים בזוג. אך מכיוון שמדובר בזוג איברים סמוכים, אזי השני תמיד במיקום אחד גדול יותר ולכן הדבר מיותר.

## פתרון שאלה 19

### א. טבלת משתנים לפונקציה f

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
n	שלם	פרמטר ראשון	שלם
k	שלם	פרמטר שני	שלם
s	מחרוזת תווים	פרמטר שלישי	תווים

#### int f (int n, int k, st s)

```

/* טענת כניסה : הפונקציה מקבלת שני מספרים שלמים ומחרוזת */
/* טענת יציאה : אם ב-s יש 'aa' תוחזר מכפלת המספרים ; אם ב-s יש 'a' יוחזר סכום המספרים ; */
/* אחרת יוחזר הפרש המספרים */(המספר הראשון פחות המספר השני)
{
if (strstr (s, "aa") != NULL) /* אם 'aa' נמצאת ב-s */
return n * k;
else
if (strchr (s, 'a') != NULL) /* אם 'a' נמצאת ב-s */
return n + k;
else
return n - k; /* אחרת, החזר את הפרש המספרים */
}

```

הפונקציה הבנויה בשפת C – strstr, מחזירה את המיקום הראשון של תת-מחרוזת (הפרמטר השני) בתוך מחרוזת (הפרמטר הראשון) – אם היא נמצאת. אם היא לא נמצאת היא מחזירה NULL. נוח מאד להשתמש בפונקציה זו כאן וקל יותר מאשר לבצע לולאת סריקה לאיתור תת-המחרוזת. בהתאם לתוצאה שהפונקציה מחזירה מתקבלת ההחלטה אילו ערכים להחזיר.



### מספר פתרונות



**פתרון נוסף:** להלן גרסת פתרון לפונקציה ללא שימוש בפונקציה strstr, אלא בסריקת תווי המחרוזת בלולאה.

### משתנים נוספים לפונקציה f

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
pos	שלם	מיקום תו תורן במחרוזת s	שלם
posaa	שלם	מיקום ראשון של תת מחרוזת 'aa' במחרוזת s	שלם
posa	שלם	מיקום ראשון של תת מחרוזת 'a' במחרוזת s	שלם

#### int f (int n, int k, st s)

```

/* טענת כניסה : הפונקציה מקבלת שני מספרים שלמים ומחרוזת */
/* טענת יציאה : אם ב-s יש 'aa' תוחזר מכפלת המספרים ; אם ב-s יש 'a' יוחזר סכום המספרים ; */
/* אחרת יוחזר הפרש המספרים. */

```

```

{
    int pos = 0;
    int posa = 0;
    int posaa = 0;
    while ( ( pos < s.Length) && ( posaa != 1) ) /* כל עוד 'a' לא נמצא ואין חריגה מן המחרוזת */
    {
        if ( s [pos] == 'a' ) /* 'aa' מציאת 'a' או מציאת 'aa' */
            if ( ( pos < s.Length) && ( s [pos + 1] == 'a') )
                posaa = 1;
            else
                posa = 1;
        pos ++;
    }
    if ( posaa == 1 )
        return n * k; /* 'aa' הופיע */
    else
        if ( posa == 1 ) /* רק 'a' הופיע */
            return n + k;
        else
            return n - k; /* 'a' אין */
}

```

ב. פתוח אלגוריתם למערך דו-מימדי בגודל 25\*35

(i) משתנים עיקריים לפתרון הבעיה

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
m	מערך דו-מימדי	איחסון הנתונים הנדרשים	מציינים : [0..24, 0..34]
p	שלם	מספר הנתונים החיוביים שנוצרו במערך m	שלם חיובי או 0
n	שלם	מספר הנתונים השליליים שנוצרו במערך m	שלם חיובי או 0

(ii) טבלת תת-משימות

שם תת-המשימה	סוג הפעולה	מטרת המשימה
int f (int *n, int *k, st s)	פונקציה המחזירה ערך שלם	הפונקציה מקבלת שני מספרים שלמים n, k ומחרוזת s, ומחזירה את מכפלתם אם 'aa' נמצאת ב-s, את סכומם אם 'a' נמצאת ב-s, ואת הפרשם אחרת.
void builtMat (mat m)	פונקציה	בונה את מערך m, בעזרת הפונקציה f. עבור כל זוג מציינים של איבר במערך תקלוט מחרוזת ותקבע את ערך האיבר על פי חישוב הפונקציה f על המציינים והמחרוזת.
void matPosNeg (mat m, int *positive, int *negative)	פונקציה	מקבלת את מערך m ומונה ומחזירה כמה מערכיו חיוביים ב-positive, וכמה מערכיו שליליים ב-negative.
תכנית ראשית	תכנית ראשית	1. מזמנת את בניית המערך. 2. מזמנת את חישוב מספר הנתונים החיוביים והשליליים ב m. 3. מדפיסה את המונים שהתקבלו.

כאשר ניתנת בעיה פתוחה או פתוחה בחלקה ויש צורך לחלק לתת-משימות, מוטב להגדיר תת-משימה לכל תהליך עיבוד בבעיה. חלוקה כזו נכונה יותר בתהליך פתרון בעיות וגם מאפשרת להתמודד עם פיתוח פתרון פשוט יותר עבור כל תת-משימה.



### (iii) מימוש הפתרון

// q19t01.c

```
typedef int mat[25][35];
typedef char st[20];
```

```
int f (int n, int k, st s)
/* הפונקציה מסעיף א */
```

#### טבלת משתנים לפונקציה builtMat

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
m	מערך דו-מימדי	איחסון הנתונים הנדרשים	מציינים : [0..24, 0..34]
line	שלם	מציין מספר שורה במערך m	0..24
col	שלם	מציין מספר עמודה במערך m	0..34
s	מחרוזת	מחרוזת קלט תורנית עבור כל איבר במערך m	תווים

```
void builtMat (mat m)
```

```
/* טענת כניסה : הפונקציה מקבלת מערך דו-מימדי m
/* טענת יציאה : הפונקציה קובעת את ערכו של כל איבר ב- m, על פי הפונקציה f
/* בהתאם למצייני האיבר ומחרוזת הקלט לכל איבר
*/
{
    int line, col;
    st s;
    for (line=0; line< 25; line++) /* col ובעמודה line בשורה
    {
        for ( col = 0; col < 35; col ++ )
        {
            Console.WriteLine ("Enter string: ");
            s = Console.ReadLine(); /* קלוט מחרוזת s */
            m [line][col] = f (line, col, s); /* חשב את הפונקציה f והשם באיבר המערך התורן */
        }
    }
}
```

#### טבלת משתנים לפונקציה matPosNeg

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
m	מערך דו-מימדי	איחסון הנתונים הנדרשים	מציינים : [0..24,0..34]
positive	שלם	מונה מספר הערכים החיוביים במערך m	שלם
negative	שלם	מונה מספר הערכים השליליים במערך m	שלם
line	שלם	מציין מספר שורה במערך m	0..24
col	שלם	מציין מספר עמודה במערך m	0..34

### void matPosNeg (mat m, int \*positive, int \*negative)

```
/* טענת כניסה : הפונקציה מקבלת מערך דו-מימדי m */
/* טענת יציאה : הפונקציה מונה ומחזירה את מספר הערכים החיוביים ב- positive */
/* ואת מספר הערכים השליליים ב- negative */
{
    int line, col;
    *positive = 0; /* אתחול המונים ל- 0 */
    *negative = 0;
    for ( line = 0; line < 25; line ++ ) /* סריקת המערך הדו-מימדי */
    {
        for ( col = 0; col < 35; col ++ )
        {
            if ( m [line][col] > 0 ) /* בדיקה וקידום של המונה המתאים */
                (*positive) ++;
            else
                if ( m [line][col] < 0 )
                    (*negative) ++;
        }
    }
}
```

הפרמטרים של המונים חייבים להיות פרמטרים של משתנה אחרת ערכיהם לא יוחזרו. מאחר ויש להחזיר שני ערכים לא ניתן להגדיר כאן פונקציה המחזירה ערך ויש צורך להשתמש בפונקציה עם פרמטרים של משתנה.

אין לשכוח שפרט מן המספרים החיוביים והמספרים השליליים יש גם מספרים השווים ל- 0. לכן חובה לנהל שני מונים ולא ניתן לקבל למשל את מספר הערכים השליליים על ידי החסרת מספר הערכים החיוביים ממספר האיברים במערך (25\*35).

לא ניתן להסתפק במבנה של if ... else .... לקידום שני המונים. חובה להכניס תנאי על נוסף אשר מאבחן את ערכי 0, אחרת הם ימנו עם השליליים (או החיוביים – בהתאם לתנאי).



### public static void main (string[] args)

```
{
    mat m;
    int p, n;
    builtMat (m);
    matPosNeg (m, &p, &n);
    Console.Write ("The number of negative elements is: "+n);
    Console.Write ("The number of positive elements is: "+p);
}
```

## פתרון שאלה 20

### א. טבלת משתנים לפונקציה pay

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
room	שלם	מספר חדר	שלם חיובי
days	שלם	מספר ימי שהות	שלם חיובי
sum	ממשי	סך הכל הוצאות נוספות על השהות	ממשי חיובי

### void pay(int room, int days, int sum)

```

/* טענת כניסה : הפונקציה מקבלת מספר חדר */
/* טענת יציאה : הפונקציה קולטת עבור מספר החדר את מספר ימי השהות וההוצאות */
/* השוטפות לכל יום ומדפיסה את כל הפרטים ואת הסכום הכולל לתשלום */
{
    Console.WriteLine ("room number: "+room);          /* הדפסת מספר החדר */
    Console.WriteLine ("number of days: "+days);      /* הדפסת מספר ימי השהות */
    Console.WriteLine ("total pay: "+ (days* 400 + sum)); /* חישוב והדפסת סך התשלום */
    if ((days * 400 + sum) > 2000)                    /* אם התשלום מעל 2000 - הדפס ארבעה תשלומים */
        Console.WriteLine ("four payments!");
}

```

### ב. (i) משתנים עיקריים לפתרון הבעיה

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
room	שלם	מספר חדר	שלם חיובי
days	שלם	מספר ימי שהות	שלם חיובי
sum	ממשי	סך כל ההוצאות הנוספות על השהות לפי מספר חדר	ממשי חיובי

**שים** המשתנים העיקריים לפתרון הבעיה לא בהכרח יופיעו בפונקציה הראשית שלה, ובמקרה זה הם מופיעים גם בשתיים מן הפונקציות, אך הם המהותיים לפתרון הבעיה.

### (ii) טבלת תת-משימות

שם תת-המשימה	סוג הפעולה	מטרת המשימה
pay (int room, int days, int sum)	פונקציה	הפונקציה מקבלת מספר חדר, מספר ימי שהות וסך כל ההוצאות הנוספות, ומדפיסה את נתוני השהות, את הסכום הכללי לתשלום, ואת אפשרות התשלום.
visitor ( int room )	פונקציה	הפונקציה מקבלת מספר חדר ומטפלת בחשבון שלו : קולטת מספר ימי שהות, קולטת לכל יום שהות את ההוצאות הנוספות וצוברת אותן, מדפיסה את כל החשבון על ידי הפונקציה pay.
תכנית ראשית	תכנית ראשית	קולטת מספרי חדרים להם נדרש חשבון ומחשבת ומדפיסה את חשבון החדר על ידי הפונקציה visitor. הקלט עד לזקיף מספר חדר 999.



מאחר וכבר הגדרנו פונקציה המטפלת בביצוע החשבון הכללי בסעיף א יש להשתמש בה. כדי להשתמש בה יש תחילה לצבור את סך ההוצאות הנוספות לכל חדר ולכן יש צורך בתת-תכנית נוספת שתקלוט ותחשב לחדר את ההוצאות הנוספות שלו. מאחר ויש לבצע עבור מספר חדרים הגיוני שמספר החדר יהיה פרמטר עבור הפונקציה visitor, מאחר והוא יקלט בתוך לולאת זקיף בתכנית הראשית. תת-המשימות לא חייבות להיות מתועדות על ידי כותרת הפונקציה המממשת אותן – ניתן בהחלט להסתפק בשלב זה בתיאור מילולי. יחד עם זאת קריא יותר להסתכל על הפתרון כאשר תת-המשימות מתועדות את המימוש שבה אחר-כך.

### (iii) מימוש הפתרון

// q20t01.c

**void pay (int room, int days, int sum)**

/\* הפונקציה מסעיף א \*/

### טבלת משתנים לפונקציה visitor

שם המשתנה	טיפוס המשתנה	הסבר/תפקיד	תחום ערכים
room	שלם	מספר חדר מתקבל כפרמטר	שלם חיובי
days	שלם	מספר ימי שהות - קלט	שלם חיובי
sum	שלם	סך הכל הוצאות נוספות על השהות	שלם חיובי
i	שלם	מציין את המספר הסידורי של יום השהות המטופל	1..days
exp	ממשי	סכום הוצאה אחת	ממשי

**void visitor (int \*room)**

```
/* טענת כניסה : הפונקציה מקבלת מספר חדר */
/* טענת יציאה : הפונקציה קולטת את מספר ימי השהות ואת ההוצאות השוטפות */
/* לכל יום ומביאה להדפסת כל פרטי השהות, ההוצאות ותנאי התשלום */
{
    int days, sum, i, exp;
    Console.WriteLine ("Enter number of days: "); /* קליטת מספר ימי השהות בחדר */
    days = int.Parse(Console.ReadLine());
    sum = 0; /* אתחול סכום ההוצאות הנוספות ל- 0 */
    for (i = 0; i < days; i++) /* לולאה לקליטת הוצאות נוספות לכל יום */
    {
        Console.WriteLine ("Enter expenses for day {0}: ", i + 1);
        exp = int.Parse(Console.ReadLine()); /* קליטת הוצאה אחת מתוך יום מספר i */
        while (exp != 0) /* לולאה לקליטת הוצאות נוספות ליום מסויים עד זקיף 0 */
        {
            sum += exp; /* צבירת ההוצאות */
            Console.WriteLine ("Enter expenses for day {0}:", i + 1);
            exp = int.Parse(Console.ReadLine()); /* קליטת הוצאה אחת מתוך יום מספר i */
        }
    }
    pay (*room, days, sum); /* זימון הפונקציה pay עם הסכום המצטבר לצורך הדפסה */
}
```



מאחר והקלט הוא של הוצאה בודדת בכל יום, ועבור כל יום יש זקיף 0, יש לולאה כפולה. הלולאה החיצונית עבור הימים, והלולאה הפנימית עבור כל יום. למרות הלולאה הכפולה הסכום המצטבר הוא כללי – אין צורך לצבור הוצאות נוספות לכל יום בנפרד. לכן יש רק אתחול אחד של הסכום ל- 0 לפני הלולאה הכפולה.



שים ♥ : שגוי לאתחל את sum ל- 0 לכל יום חדש. הצבירה היא של כלל ההוצאות ולא לכל יום בנפרד. אתחול ל- 0 בתוך הלולאה החיצונית יביא לצבירה בכל פעם של יום אחד שתמחק על ידי איפוס היום הבא.

## מספר פתרונות



**פתרון נוסף:** ניתן לנהל שני צוברים עבור ההוצאות: sum – עבור ההוצאות המצטברות הכלליות, ו- sumDay עבור ההוצאות ליום אחד. במקרה זה יש לאפס את ההוצאות היומיות עם התחלת כל יום, ולצבור אל sum את סך ההוצאה היומית. ראה בקוד הבא:

```
sum = 0; /* אתחול סכום ההוצאות הכללי ל- 0 */
for ( i = 0; i < days; i++) /* לולאה לקליטת הוצאות נוספות לכל יום */
{
    sumDay = 0;
    Console.Write ("enter expenses for day {0}: ", i );
    exp = int.Parse(Console.ReadLine()); /* קליטת הוצאה אחת מתוך יום מספר i */
    while (exp != 0) /* לולאה לקליטת הוצאות נוספות ליום מסויים עד זקיף 0 */
    {
        sumDay += exp; /* צבירת ההוצאות היומיות */
        exp = int.Parse(Console.ReadLine()); /* קליטת הוצאה אחת מתוך יום מספר i */
    }
    sum += sumDay; /* צבירת ההוצאות הכלליות */
}
```

```
public static void main (string[] args) /* התכנית הראשית */
{
    int room;
    Console.Write ("Enter room number: ");
    room = int.Parse(Console.ReadLine());
    while ( room != 999 ) /* לולאה ראשית לטיפול בלקוחות שונים על פי מספר חדר */
    {
        visitor (&room); /* זימון הפונקציה visitor לטיפול בלקוח מחדר מסוים */
        Console.WriteLine ("=====");
        Console.Write ("Enter room number: ");
        room = int.Parse(Console.ReadLine());
    }
}
```