

מדעי המחשב - 2 יחידות לימוד  
פתרון בחינת הבטחות

פרק א

שאלה 1:

```
read (a, b, c);
if (a + b = 200) or (a + c = 200) or (b + c = 200) then
    writeln (" yes ")
else
    writeln (" no ");
```

שאלה 2:

m1	m2	n	k	k≤4	s	t	s*t>n	s-t>0	פלט
0	0	8							2 3
1	1		1	T	6	4	T	T	
			2	T	2	3	F	F	
2	2		3	T	5	2	T	T	
	3		4	T	7	0	F	T	

שאלה 3:

קטע תכנית: בכל תא במערך נמצא המספר 1 או המספר 7.

```
count1 := 0;
count2 := 0;
for i := 1 to 51 do
    if arr[i] = 1 then
        count1 := count1 + 1
    else
        count2 := count2 + 1 ;
if (count1 > count2) then
    writeln (1)
else
    writeln (7);
```

אפשרות נוספת:  
ניתן לספור רק את מספר המופעים של 1 ל- count1  
ובסיום הלולאה לחשב: count2 := 51 - count1;

שאלה 4:

a	b	$a \neq b$	num1	num2	פלט
2	3	T	23	32	23
					32
					55

ב. 3, 3 (כל שני מספרים זהים).  
 ג. (i) 3, 1  
 (ii) 4, 4

שאלה 5:

```

for i := 1 to 38 do
  begin
    read (a, b);
    writeln (a + b);
    if b = 0 then
      writeln (' indivisible ')
    else
      writeln( trunc (a / b));
  end;

```

הערה : div מחזיר את מנת החילוק, ולכן ניתן גם להדפיס את : a div b .

פרק ב

## שאלה 6:

```
{
    קלט : מחיר של כל אחד מפריטי הלבוש שנקנו בחנות.
    פלט : גובה ההנחה (אם קיימת) והמחיר לאחר ההנחה.
    האם מגיע לקונה שובר הנחה לקניה הבאה
}
```

**Program Targil6;****Var**count, price, sum, n, discount : **integer**;**Begin**

count := 0;

sum := 0;

write (' → מחיר הפריט הראשון ?');

read (price);

**while** price <> 0 **do****begin**

sum := sum + price;

count := count + 1 ;

write (' → מחיר הפריט הבא ?');

read (price);

**end**;n := sum **div** 800 ;**if** n <> 0 **then**

discount := 50 \* n

**else**

discount := 0;

writeln (discount, ' : גובה ההנחה');

sum := sum - discount ;

writeln (sum, ' : מחיר הקניה');

**if** count >= 4 **then**

writeln (' זכאי לשובר הנחה');

**End.**

שאלה 7:

	1	2	3	4	5	6	7	8	9	10	11
arr	6	7	8	11	2	0	2	3	17	7	6

sum	g	k	i	$i \geq 0$	I arr[i]	II arr[g-i]	$I == II$
0	12	6					
1			5	T	2	2	T
			4	T	11	3	F
			3	T	8	17	F
2			2	T	7	7	T
3			1	T	6	6	T

בסיום הביצוע ערכו של sum יהיה: 3

	1	2	3	4	5	6	7	8	9	10	11
a	1	2	3	4	5	6	1	2	3	4	5

ב.

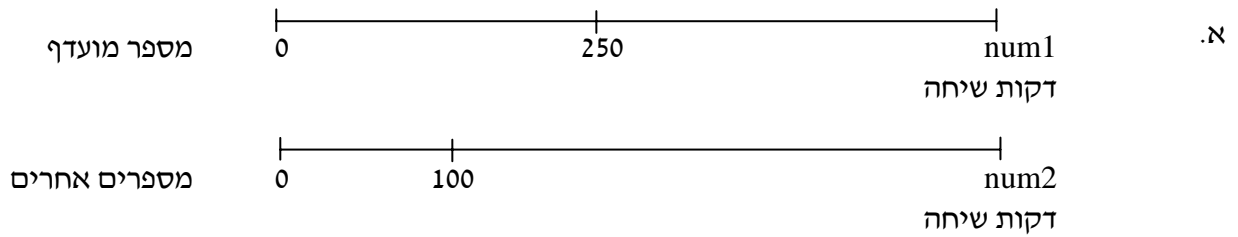
כל מערך שאין בו סימטריה.

ג.

sum	g	k	i	$i \leq g-1$	I arr[i]	II arr[g-i]	$I == II$
0	12	6					
1			7	T	2	2	T
			8	T	3	11	F
			9	T	17	8	F
2			10	T	7	7	T
3			11	T	6	6	T

ערכו של sum יהיה: 3

שאלה 8:



{ טענת כניסה : מספר דקות השיחה למספר מועדף num1  
 מספר דקות שיחה למספרים אחרים num2  
 } טענת יציאה : הסכום בשקלים שיש לשלם

**function** pay (num1, num2 : integer) : real ;

**var**

sum : real ;

**begin**

sum := 0;

**if** num1 > 250 **then**

sum := sum + (num1 - 250) \* 0.2;

**if** num2 > 100 **then**

sum := sum + (num2 - 100) \* 0.3;

pay := sum;

**end;**

ב. קטע התכנית :

write (' → מספר דקות שיחה למספר מועדף ? ');

read (num1);

**while** num1 >= 0 **do**

**begin**

write (' → מספר דקות שיחה למספרים אחרים ? ');

read (num2);

toPay := pay (num1 , num2);

writeln (toPay:5:1, ' : הסכום לתשלום');

write (' → מספר דקות שיחה למספר מועדף ? ');

read (num1);

**end;**

פרק ג

## שאלה 9:

**Program** Targile9;

**Type** mat\_Type = **array** [1..17, 1..17] of **integer** ;

**Var**

m : mat\_type;

i, j, count : **integer** ;

{ טענת כניסה: מערך דו-ממדי m, מספר שורה line, ומספר עמודה col  
טענת יציאה: מוחזר 1 אם האיבר שבמקום זה הוא איבר צומת. }

**Function** nodeItem (m : mat\_type ; line, col : **integer**) : **integer** ;

**begin**

**if** sumOfLine (m, line) = sumOfColumn (m, col) **then**

nodeItem := 1

**else**

nodeItem := 0 ;

**end;**

{ טענת כניסה: מטריצה m ושורה line  
טענת יציאה: מוחזר סכום השורה }

**Function** sumOfLine (m : mat\_type ; line : **integer**) : **integer**;

**var** i, sum : **integer** ;

**begin**

sum := 0;

**for** i := 1 **to** 17 **do**

sum := sum + m[line , i];

sumOfLine := sum;

**end ;**

{ טענת כניסה: מטריצה m ועמודה col  
טענת יציאה: מוחזר סכום העמודה. }

**Function** sumOfColumn (m : mat\_type ; col : **integer**) : **integer** ;

**var** i, sum : **integer** ;

**begin**

sum := 0;

**for** i := 1 **to** 17 **do**

sum := sum + m[i , col];

sumOfColumn := sum;

**end;**

```
Begin { main program }
  int count = 0;
  mat_type mat;
  for i := 1 to 17 do
    for j := 1 to 17 do
      if nodeItem (mat, i, j) = 1 then
        count := count + 1 ;
  writeln (count , ' מספר איברי הצומת שנמצאו : ');
End .
```

Program Targil10;

שאלה 10:

Var

i, min, first, second, time : integer ;

{ טענת כניסה: שני מספרים שלמים first ו- second בני 4 ספרות כל אחד, המייצגים שעות.  
טענת יציאה: מספר הדקות שעברו בין first ל- second.  
הנחה:  $1200 \leq \text{first} < \text{second} \leq 2359$  }

Function elapsedTime (first, second : integer) : integer ;

var

firstMinute, firstHour, secondMinute, secondHour, hours, minutes : integer ;

begin

firstMinute := first mod 100; { מספר הדקות }

firstHour := first div 100; { מספר השעות }

secondMinute := second mod 100; { מספר הדקות }

secondHour := second div 100; { מספר השעות }

hours := secondHour - firstHour ;

(\*) if secondMinute >= firstMinute then  
minutes := secondMinute - firstMinute

else

begin

minutes := (60 + secondMinute) - firstMinute;

hours = hours - 1;

end ;

elapsedTime := (hours \* 60 + minutes) ;

end ;

Begin

{ main program }

min := 12\*60 ;

for i := 1 to 127 do

begin

write ('שעת ההתחלה →');

read (first);

write ('שעת הסיום →');

read (second);

time := elapsedTime (first, second);

writeln (time, ' : זמן הריצה');

if time < min then

min := time;

end ;

writeln (min, ' : זמן מינימלי');

End.

דרך פשוטה יותר לחשוב מספר הדקות  
(לפי הצעתה של עפרה בן-ארי):  
החזר:

(secondHour\*60 + secondMinute) -  
(firstHour\*60 + firstMinute);

(\*) למעשה, כדי לחשב את מספר הדקות, מספיק לחשב:

minutes := secondMinute - firstMinute;

החזר: (hours \* 60 + minutes)

(אם דקות ההתחלה קטן מדקות הסיום יהיה minutes מספר שלילי)