

מדעי המחשב - 2 יחידות לימוד
פתרון בחינת הבטלות

פרק א

שאלה 1:

k	num	i	x	x זוגי?	פלט
1	0				3126
	2	1	2	T	
10	12	2	1	F	
100	312	3	3	F	
	3126	4	6	T	

קלט: 2, 1, 3, 6 ⇒

שאלה 2:

	x	y	$x > 0$	פלט
א.	10	4	T	-2
	6		T	
	2		T	
	-2			
ב.	-10	4	F	-10
ג.	10	4	T	6
	6			

סעיף ב: כל מספר שלילי או 0 מתקבל.
(אין צורך בטבלת מעקב עבור סעיף זה).

שאלה 3:

קטע תכנית: סכום ומספר המספרים התלת ספרתיים במערך a שגודלו 62.

```

sum := 0;      count := 0;
for i := 1 to 62 do
    if (a[i] >= 100) and (a[i] <= 999) then      {if (a[i] > 99) and (a[i] < 1000) then : אפשר גם}
        begin
            sum := sum + a[i];
            if a[i] > 248 then
                count := count + 1;
        end;
writeln (sum, ' : סכום המספרים התלת ספרתיים ');
writeln (' : היו ', count, ' מספרים תלת-ספרתיים גדולים מ-248 ');

```

שאלה 4:

השגיאה: בשורה האחרונה: מודפס סכום המספרים חלקי המספר האחרון.
(השגיאה חמורה במיוחד אם הערך האחרון שנקלט ל- num הוא 0).

התיקון: יש לחלק במספר המספרים: `writeln (sum / count);`

שאלה 5:

א. משפט הזימון תקין. הפלט 0

ב. משפט הזימון אינו תקין. חסרים הפרמטרים שיישלחו לפונקציה.

ג. משפט הזימון תקין:

תכנית		פונקציה			ערך מוחזר / פלט
m	i	a	b	$b-a > 5$	
7	1	1	7	T	1
	2	2	7	F	0
	3	3	7	F	0

פרק בשאלה 6:

קלט: מספר הכרטיסים להגרלה, למופע ולמזנון שנרכשו ביריד.
 פלט: מספר המשתתפים ביריד, כמה כרטיסים מכל סוג נמכרו, ומספר כרטיסי ההגרלה שניתנו חינם.

```
program yarid;
```

```
var
```

```
  hagT, mufT, mizT : integer;      { מספר הכרטיסים שנרכשו להגרלה, מופע ומזנון }
  sHag, sMuf, sMiz : integer;      { צוברים לכל סוגי הכרטיסים }
  mone, freeHagT : integer;        { מונה מספר המבקרים, מספר כרטיסי החינם }
```

```
begin
```

```
  sHag := 0;      sMuf := 0;      sMiz := 0;
  mone := 0;      freeHagT := 0;
```

```
  write (' → מספר הכרטיסים להגרלה ? '); read (hagT);
```

```
  while (hagT <> -1) do
```

```
    begin
```

```
      write (' → מספר הכרטיסים למופע ? ');      read (mufT);
```

```
      write (' → מספר הכרטיסים למזנון ? ');      read (mizT);
```

```
      mone := mone + 1;      { ספירת מספר המבקרים }
```

```
      sHag := sHag + hagT;   { ספירת מספר כרטיסי ההגרלה }
```

```
      sMuf := sMuf + mufT;   { ספירת מספר הכרטיסים למופע }
```

```
      sMiz := sMiz + mizT;   { ספירת הכרטיסים למזנון }
```

```
      if (hagT > 10) then
```

```
        freeHagT := freeHagT + 1; { בדיקת זכאות לכרטיס הגרלה חינם }
```

```
      write (' → מספר הכרטיסים להגרלה ? ');      read (hagT);
```

```
    end;
```

```
  writeln (' היו ', mone, ' מבקרים ביריד');
```

```
  writeln (' (נמכרו ', sHag, ' כרטיסי הגרלה');
```

```
  writeln (' (נמכרו ', sMuf, ' כרטיסים למופע');
```

```
  writeln (' (נמכרו ', sMiz, ' כרטיסים למזנון');
```

```
  writeln (' (חולקו ', freeHagT, ' כרטיסי חינם להגרלה');
```

```
end.
```

לפינה 7:

```

{--- קלט: מספרים דו-ספרתיים למערך ---}
{--- פלט: מספר המספרים שבהם סכום הספרות שבין ספרת האחדות וספרת העשרות גדול מ-15 ---}
program bigArrSum ;
var
  arr : array [1..23] of integer;
  sum, i, count : integer;
{-----}
{ ט.כניסה: מספר דו-ספרתי שלם וחיובי שספרותיו שונות זו מזו. }
{ ט.יציאה: סכום המספרים החד-ספרתיים שבין ספרות המספר. }
function bigSum ( a : integer) : integer;
var
  d1, d10 : integer;           { ספרות המספר }
  i, sum : integer;
begin
  sum := 0;
  d1 := num mod 10;
  d10 := num div 10;
  if d1 < d10 then
    for i := d1 to d10 do
      sum := sum + i
    else
      for i := d1 downto d10 do
        sum := sum + i;
      bigSum := sum;
end;
{---- main program ----}
begin
  { כאן יבוא קטע תכנית המבצע קלט למערך הנתון }
  count := 0;
  for i := 1 to 23 do
    begin
      sum := bigSum (arr[i]);
      if sum > 15 then
        count := count + 1;
      end.
    writeln ('(נמצאו ', count, ' מספרים המקיימים את התנאי ');
end.

```

דרכים אחרות לחישוב סכום הספרות:

- if (d1 > d10) then swap (d1, d10);
for i := d1 to d10 do sum := sum + i;
swap (x,y) = פרוצדורה להחלפה בין x ו-y.
(יש לכתוב פרוצדורה זו בתכנית).
- if d1 < d10 then
for i := d1 to d10 do sum := sum + i
else
for i := d10 to d1 do sum := sum + i;

בשאלה ביקשו רק קטע תכנית,
לכן מספיק לכתוב קטע זה.

לצוה 8:

b

	1	2	3	4	5	6	7	8	9	10
	0	2	4	4	5	2	8	8	10	10

a

	1	2	3	4	5
	2	17	3	6	20
		-1	-1	-1	

n	count	i	$a[i] \neq b[i*2-1] + b[i*2]$	קיום התנאי	פלט
5	0				3
	1	1	$2 \neq 0 + 2$	F	
		2	$17 \neq 4 + 4$	T	
		3	$3 \neq 5 + 2$	T	
		4	$6 \neq 8 + 8$	T	
	2	5	$20 \neq 10 + 10$	F	

ב. הסבר: מכיוון שמודפס הערך של $\{n - \text{count}\}$ והמונה גדל ב-1 עבור כל ערך של מערך a שאינו שונה מסכום 2 האיברים במערך b, הרי שאיברי מערך a צריכים להיות שווים לסכום שני התאים המתאימים במערך b.

a

	1	2	3	4	5
	2	8	7	16	20

פרק ג

לסוף 9:

```
{--- קלט: מערך דו-ממדי (מטריצה) שאיבריו הם 0 או 1 (בלבד) ---}
{--- פלט: מספר האיברים ה"מוקפים" במערך. ---}
{--- איבר "מוקף" הוא איבר שערכו 1 ואורך השרשרת בשורה ובעמודה שלו שווים ---}
```

```
program surroundedMatrixItem;
```

```
const N = 10;
```

```
type mat_type = array [1..N,1..N] of integer;
```

```
var m : mat_type ;
    i, j, count : integer;
```

```
{-----}
{--- מוחזר אורך השרשרת בשורה r במטריצה m ---}
function rowChain (m : mat_type ; r : integer) : integer;
```

```
var j, count : integer;
```

```
begin
```

```
    count := 0;
```

```
    for j := 1 to N do
```

```
        count := count + m[r, j];
```

```
    rowChain := count;
```

```
end;
```

הנחת התכנית: איברי המערך הם 0 או 1
ויש רק שרשרת אחת בכל שורה או עמודה,
ולכן מספיק לסכם את איברי השורה/עמודה.

```
{-----}
{--- מוחזר אורך השרשרת בעמודה c במטריצה m ---}
function colChain (m : mat_type ; c : integer) : integer;
```

```
var i, count : integer;
```

```
begin
```

```
    count := 0;
```

```
    for i := 1 to N do
```

```
        count := count + m[i, c];
```

```
    colChain := count;
```

```
end;
```

```
{-----}
{--- טענת כניסה: מטריצה m, שורה i ועמודה j ---}
{--- טענת יציאה: 1 אם האיבר במקום זה הוא איבר "מוקף" ו-0 אחרת. ---}
{--- איבר מוקף = ערכו של האיבר שבתא (i, j) 1 ואורך שרשרת השורה שלו. ---}
{--- שווה לאורך שרשרת העמודה שלו ---}
```

```
function surroundedItem (m : mat_type ; i, j : integer) : integer ;
```

```
begin
```

```
    if (m[i, j] = 1) and (rowChain (m, i) = colChain (m, j)) then
```

```
        surroundedItem := 1
```

```
    else surroundedItem := 0;
```

```
end;
```

```

{--- main program ---}
begin
  { כאן יבוא קטע תכנית המבצע קלט למערך הנתון }
  count := 0;
  for I := 1 to N do
    for j := 1 to N do
      count := count + surroundedItem (m, i, j);
  writeln (count, ' מספר האיברים המוקפים הוא: ');
end.

```

לצורה 10:

טבלת משתנים:

שם משתנה	טיפוס משתנה	תפקיד המשתנה
books	מערך בגודל 315 של שלמים	מערך הצוברים לסך הספרים שהוזמנו מכל סוג.
num	שלם	מספר העותקים של הספר שהזמין התלמיד.
cat	שלם (בין 1 ל- 315)	המספר הקטלוגי של הספר שהזמין התלמיד.
sum	שלם	הסכום הכולל שעל התלמיד לשלם עבור הזמנתו.
total	שלם	הסכום הכולל שעל בית הספר לשלם עבור כלל ההזמנה.

תת-פצילות:

אתחול-מערך (books) `procedure books_Init (var books : arr_type);`
 פעולה המתחלת את כל איברי המערך ל-0.

הזמנת-תלמיד (books) `procedure student_Order (var books : arr_type);`
 טענת כניסה: מערך הספרים.
 טענת יציאה: הפעולה מוסיפה לכל תא במערך הספרים, המתאים למספר הקטלוגי של הספר שהזמין התלמיד, את מספר הספרים שהוזמנו.

הצג-הזמנות (books) `procedure books_Oreders_Show (books : arr_type);`
 טענת כניסה: מערך הספרים.
 טענת יציאה: הפעולה מדפיסה לכל ספר את מספר ההזמנות שבוצעו עבורו.

סכום-סך-ההזמנות (books) `function books_Total_Order (books : arr_type) : integer;`
 טענת כניסה: מערך הספרים.
 טענת יציאה: מוחזר סכום איברי המערך מוכפל במחיר הספר (28 ש"ח).

```

{--- קלט : הזמנת הספרים לכל תלמיד ---}
{--- פלט : פירוט וסך ההזמנות לבית הספר. ---}

program books_order ;
const
    N = 620;
    BOOK_NUM = 315;
    BOOK_PRICE = 28;
type
    arr_type = array [1.. BOOK_NUM] of integer ;
var
    books : arr_type ;
    total : integer;

{-----}
{ פעולה המאתחלת את כל איברי המערך ל-0. }

procedure books_Init (var books : arr_type);
var
    i : integer;
begin
    for i := 1 to BOOK_NUM do
        books[i] := 0;
end;

{-----}
{ ט. כניסה : מערך הספרים. }
{ ט. יציאה : הפעולה מוסיפה לכל תא במערך הספרים, המתאים למספר הקטלוגי של הספר שהזמין }
{ התלמיד, את מספר הספרים שהוזמנו. }

procedure student_Order (var books : arr_type);
var
    num, cat, sum : integer;
begin
    sum := 0;
    write (' (1..', BOOK_NUM, ') מספר קטלוגי לספר → ');
    read (cat);
    while cat <> 0 do
        begin
            write (' מספר הספרים → '); read (num);
            books [cat] := books [cat] + num;
            sum := sum + num;

            write (' (1..', BOOK_NUM, ') מספר קטלוגי לספר → ');
            read (cat);
        end;
    writeln (sum * BOOK_PRICE, ' : סך ההזמנה לתלמיד ');
end;

```

```

{-----}
{
    טענת כניסה : מערך הספרים.
}
{
    טענת יציאה : הפעולה מדפיסה לכל ספר את מספר ההזמנות שבוצעו עבורו.
}

```

```
procedure books_Order_Show (books : arr_type);
```

```
var i : integer;
```

```
begin
```

```
  writeln ('פירוט ההזמנות לבית הספר :');
```

```
  writeln ('מספר קטלוגי', 'כמות ספרים', '15');
```

```
  for i := 1 to BOOK_NUM do
```

```
    writeln (books[i]:7, i:10);
```

```
end;
```



```

{-----}
{
    טענת כניסה : מערך הספרים.
}
{
    טענת יציאה : מוחזר סכום איברי המערך מוכפל במחיר הספר (28 ש"ח).
}

```

```
function books_Total_Order (books : arr_type) : integer;
```

```
var
```

```
  i, sum : integer;
```

```
begin
```

```
  sum := 0; { צובר לסכום המערך = sum }
```

```
  for i := 1 to BOOK_NUM do
```

```
    sum := sum + books [i];
```

```
  books_Total_Order := sum * BOOK_PRICE;
```

```
end;
```

```
{--- main program ---}
```

```
begin
```

```
  books_init (books);
```

```
  for i := 1 to N do
```

```
    student_Order (books);
```

```
  books_Order_Show (books);
```

```
  total := books_Total_Order (books);
```

```
  writeln (' ש"ח ', total, ' (סך הזמנת בית הספר :');
```

```
end.
```