

מדעי המחשב - 2 יחידות לימוד
פתרון בחינת הבגרות

פרק א

שאלה 1:

- א. יחזור 7
- ב. יחזור 0
- ג. טענת כניסה: 2 מספרים שלמים a ו-b. טענת יציאה: אם הראשון גדול יותר יוחזר הראשון, אחרת יוחזר ההפרש שבין שני המספרים.

שאלה 2:

pul	i	num	num>i	פלט
100.0				7.0
90.0	1	10	T	
45.0	2	1	F	
15.0	3	3	F	
7.0	4	8	T	

טבלת מעקב לקלט: 10, 1, 3, 8 →
פלט האלגוריתם: 7

הערה: עמודת בדיקת התנאי אינה חייבת להיות בטבלה. הימצאותה עוזר לבצע את המעקב.

T = True, F = False

בצעד 2.2.2 מקבל המשתנה pul תוצאה של חילוק (חילוק ממשיים) ולכן ניתן להניח כי pul הוא מטיפוס ממשי.

שאלה 3:

נדרש קטע תבנית. שם המערך: A

```
count_1 := 0 ; count_10 := 0 ;
for i := 1 to 28 do
    if A[i] < 10 then
        count_1 := count_1 + 1
    else
        count_10 := count_10 + 1 ;
if count_10 > count_1 then
    writeln (' דו ספרתיים ')
else
    writeln (' חד ספרתיים ');
```

ניתן לשאול על מנת החלוקה ב-10:

```
if A[i] div 10 = 0 then
    count_1 := count_1 + 1
else
    count_10 := count_10 + 1 ;
```

```
if counter > (28-counter) then
    writeln (' דו ספרתיים ');
else
    writeln (' חד ספרתיים ');
```

ניתן להשתמש במונה אחד שישפור למשל רק את הדו ספרתיים, ולבדוק:

שאלה 4:

דרך אחת	דרך אחרת	
	writeln (st1) ;	א.
for i := 1 to 4 do write (st1[i]) ;	st2 := copy (st1, 1, 4) ; writeln (st2) ;	ב.
found := false ; for i := 1 to 10 do if st1[i] = 'Y' then found := true ; if found then writeln ('yes') else writeln ('no') ;	p := pos ('Y', st1) ; if p <> 0 then writeln ('yes') else writeln ('no') ;	ג.

שאלה 5:

	1	2	3	4
1	5	3	7	4
2	1 2	4 2	9 2	16 2
3	7 8	9	8	8

t	s
1	3
2	
3	
4	

פרק ב

שאלה 6:

```

sum := 0 ;
write ('(M/F) ? מין המשתתף → ');
read (ch) ;
if ch = 'M' then
  for i := 1 to 10 do
    begin
      write (' ? ', i, ' הישג בתחרות → ');
      read (score) ;
      sum := sum + score ;
    end
else
  for i := 1 to 7 do
    begin
      write (' ? ', i, ' הישג בתחרות → ');
      read (score) ;
      sum := sum + score ;
    end ;
writeln (sum, ' הציון הסופי : ');

```

הערות:

(1) ניתן היה לחסוך בכתיבה, ולבצע:
 if (ch = 'M') then
 limit := 10
 else
 limit := 7 ;
 ולכתוב לולאה אחת שמתבצעת מ-1 ועד limit.

(2) ניתן לפתור את השאלה באמצעות שימוש בפונקציה score המקבלת כפרמטר את מספר המשחקים (7 או 10) ומחזירה את הסכום.
 לולאת הקלט תתבצע בתוך הפונקציה.

שאלה 7:

```

{--- החלפת ערכי המערך בסכום המספרים המתחלקים ב-3 בין 1 לתוכן התא ---}
program numbers_in_array ;
type arr_type = array [1..19] of integer ;
{--- טענת כניסה: מספר שלם וחיובי n הקטן מ-100 ---}
{--- טענת יציאה: מוחזר סכום כל המספרים בין 1 ל-n (לא כולל) המתחלקים ב-3 ללא שארית. ---}
function sum_mod_3 (n : integer) : integer ;
var i, sum : integer ;
begin
  sum := 0 ;
  for i := 1 to n-1 do
    if i mod 3 = 0 then
      sum := sum + i ;
  sum_mod_3 := sum ;
end ;
procedure arr_kelet (var arr : arr_type) ;
begin
  קלט של מספרים חיוביים הקטנים מ-100 למערך הנתון
end ;

```

```

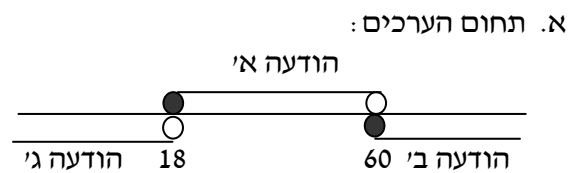
var   change : arr_type ;
      i : integer ;

begin
  arr_kelet (change) ;
  for i := 1 to 19 do
    change [i] := sum_mod_3 (change[i]) ;
  end .

```

שאלה 8:

נימוק	a	פלט
$18 \leq a < 60$	50	הודעה א
$60 \leq a$	100	הודעה ב
$a < 18$	10	הודעה ג



ב.

```

if a < 18 then writeln ('הודעה ג')
else if a < 60 then writeln ('הודעה א')
else writeln ('הודעה ב');

```

פרק ג

שאלה 9:

משתנים עיקריים:

שם המשתנה	טיפוס המשתנה	תפקיד המשתנה
T	מערך בגודל 100 של מספרים בתחום 0 - 50	מערך הטיולים. מציין המערך = מספר הטיול. תוכן המערך = מספר הנרשמים לטיול.
dd	מספרי שלם	יום. למעשה אין חשיבות ליום, כי מדובר בראשון בחודש.
mm	מספרי שלם (בתחום 1 - 12)	חודש.
yy	מספרי שלם	שנה

תת-משימות:

- הגדרת המערך ואתחולו (איפוס כל האיברים).
- בדיקת-תקינות-תאריך: פעולה המקבלת חודש ושנה ומחזירה "אמת" אם התאריך מאוחר ל-1.8.2004 ו-"שקר" אחרת.
- טיולים-ללא-ביקוש: פעולה המקבלת מערך טיולים ומדפיסה את כל הטיולים שמספר הנרשמים להם הוא 0.
- קלט-מספר-טיול: פעולה המחזירה מספר טיול תקין (בתחום 1 עד 100).

```
{--- "שלם וסעי" סוכנות נסיעות ---}
```

```
program traveling_agency ;
```

```
const N = 100;
```

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

```
var trip_num, dd, mm, yy : integer ;
```

```
{--- פעולה המקבלת חודש ושנה ומחזירה "אמת" אם התאריך מאוחר (או שווה) ל- 1.8.2004 ---}
```

```
{--- ו-"שקר" אחרת. הנחה: חודש ושנה תקינים. ---}
```

```
function valid_date (month , year : integer) : boolean ;
```

```
begin
```

```
if (year > 2004) or ((year = 2004) and (month >= 8)) then
```

```
    valid_date := TRUE
```

```
else valid_date := FALSE;
```

```
end ;
```

```
valid_date := (year > 2004) or ((year = 2004) and (month >= 8)) ;
```

{--- פעולה המקבלת מערך טיולים ומדפיסה את מספרי הטיולים שיש להם 0 נרשמים ---}

```
procedure no_demand_trip (T : arr_type) ;  
var i : integer ;  
begin  
  writeln (' מספרי הטיולים שלא היו להם נרשמים ');  
  for i := 1 to N do  
    if (T[i] = 0) write (i:5) ;  
  writeln ;  
end ;
```

{--- פעולה המחזירה מספר טיול תקין (בין 1 ל-N כולל) ---}

```
function get_num : integer ;  
var num : integer ;  
begin  
  repeat  
    write (' (, N, '-ל-1 בין) → הקש מספר טיול ');  
    read (num) ;  
  until ((num >= 1) and (num <= N)) or (num = 999) ;  
  get_num := num ;  
end ;
```

{--- פעולה המקבלת מערך ומאפסת את איבריו ---}

```
procedure arr_init (var T : arr_type);  
var i : integer ;  
begin  
  for i := 1 to N do  
    T [i] := 0 ;  
end ;
```

Begin

```
  arr_init (T) ;  
  trip_num := get_num ;  
  while trip_num <> 999 do  
    begin  
      write (' → הקש תאריך דרכון (יום, חודש, שנה) ');  
      read (dd, mm, yy) ;  
      if valid_date (mm, yy) and ( T[trip_num] < 50) then  
        T[trip_num] := T[trip_num] + 1 ;  
      trip_num := get_num () ;  
    end ;  
  no_demand_trip (T) ;
```

End.

לאלף 10:

```

{--- מטריצה "רביעייה k" ---}
program SquareK_Matrix ;
type mat_type = array [1..12, 1..12] of integer ;
var M : mat_type ;
    i : integer ;

procedure mat_kelet (M : mat_type) ;
begin
    המערך נתון. אין צורך במימוש הפעולה בבחינה.
end ;

{--- פעולה המקבלת מטריצה ומספר שלם (בתחום 1 עד 30) ---}
{--- ומחזירה "אמת" אם קיימת רבעיית תאים (2x2) שערכם k ---}
{--- ו-"שקר" אחרת. ---}
function square_k (M : mat_type; k : integer) : boolean ;
var i, j : integer ;
    found : Boolean ;
begin
    found = FALSE ;
    for i := 1 to 11 do
        for j := 1 to 11 do
            if M[i,j] = k then
                if (M[i,j+1] = k) and (M[i+1,j] = k) and (M[i+1,j+1] = k) then
                    found := TRUE ;
                square_k := found ;
            end ;
        end ;
    end ;
Begin
    i := 30 ;
    mat_kelet (M) ;
    while (i > 0) and not square_k (M, i) do
        i := i - 1 ;
    end ;
    if i = 0 then
        writeln (' אין רביעייה k ');
    else
        writeln (' נמצאה רביעייה i, ' (ערך מקסימאלי) ');
    end ;
End .

```

i, j	i, j+1
i+1, j	i+1, j+1

ניתן לייעל את הפונקציה ולעצור את החיפוש כשנמצאה רביעיית k :

```
function square_k (M : mat_type; k : integer) : boolean ;
var   i, j : integer ;
      found : Boolean ;
begin
  found = FALSE ;
  i := 1 ;
  while ( i < 12 ) and not found do           {--- 0 ≤ i ≤ 11 ---}
  begin
    j := 1 ;
    while ( j < 12 ) and not found do         {--- 0 ≤ j ≤ 11 ---}
    begin
      if M[i, j] = k then
        if (M[i, j+1] = k) and (M[i+1, j] = k) and (M[i+1, j+1] = k) then
          found := TRUE ;
        j := j + 1 ;
      end ;
      i := i + 1 ;
    end ;
    square_k := found ;
  end ;
end ;
```