

מדעי המחשב – 2 יחידות לימוד

פתרון בחינת הבטחות

פרק א

שאלה 1:

number	פלט
3	3
-7 7	7
-2 2	2
4	4
1	1
0	

א. טבלת מעקב לקלט: 0, 1, 4, -2, -7, 3 →

ב. האלגוריתם מדפיס את הערך המוחלט של המספרים שנקלטו.

שאלה 2:

טענת כניסה: הפונקציה מקבלת שני מספרים שלמים ל- a ול- b
 טענת יציאה: הפונקציה מחזירה את הממוצע שלהם.

א. function **mean** (a, b : integer) : real ;

```
begin
    mean := (a+b)/2 ;
end ;
```

ב. הקריאה לפונקציה: writeln (mean (x,y)) ;

שאלה 3:

א. להלן פתרונות שונים (התלמיד היה צריך להריץ פעם אחת בלבד)

a	b	I a<b	II a<100	I or II	פלט
150	50	שקר	שקר	שקר	שקר
70	50	שקר	אמת	אמת	אמת
150	250	אמת	שקר	אמת	אמת
50	70	אמת	אמת	אמת	אמת

ב. אם אין סוגריים הקובעים אחרת, אזי and קודם ל-or.

הקשר הלוגי האחרון לביצוע הוא or, ולכן מספיק שאחד הביטויים יהיה אמת כדי שערך כל הביטוי יהיה אמת.

$$x = -2, y = 5, z = 13$$

I	II	III	II and III	תוצאה
$z > x$	$x < 0$	$z - y > 9$		
אמת	אמת	שקר	שקר	אמת

or

שאלה 4:

התלמיד נדרש לכתוב קטע תכנית בלבד. בפתרון מוצגת תכנית מלאה.

דרך א: ניתן לכתוב קטע תכנית שיפסיק את החיפוש ברגע שמצא מחרוזת זהה. במקרה זה תהיה התשובה בסעיף ב: "ידיפס את מיקומה של המחרוזת הראשונה - במקום 3".

{ --- מילוי מערך של מחרוזות במחרוזות --- }
 { --- קלט מחרוזות והדפסת מיקומה במערך --- }

```

program tar_4 ;
const N = 27 ;
type
    string_type = string[100] ;
    arr_type = array [1..N] of string_type ;
var
    a : arr_type ;
    st : string_type ;
    found : boolean ;
    i : integer ;
{-----}
procedure arr_kelet (var a:arr_type) ;
var
    i : integer ;
begin
    writeln (' Type ',N,' strings : ' ) ;
    for i := 1 to N do
        readln (a[i]) ;
end;
{-----}
    
```

```

procedure arr_pelet (a:arr_type) ;
var
    i : integer ;
begin
    for i := 1 to N do
        writeln (a[i]) ;
end;
{-----}
Begin
    arr_kelet (a) ;
    arr_pelet (a) ;
    write (' Type a new strings ---> ');
    readln (st) ;
    found := FALSE ;
    i := 1 ;
    while (not found) and (i <= N) do
        begin
            if st = a[i] then
                found := TRUE ;
            inc (i) ;
        end ;
    writeln (st,' was fount at place: ',i-1) ;
End.

```

דרך ב: ניתן לכתוב קטע תכנית שידפיס את מיקום המחרוזת בכל פעם שמוצא מחרוזת זהה, במקרה זה תהיה התשובה בסעיף ב: "יידפיס את מיקומן של כל המחרוזות הזהות - 20,3".

```

Begin
:
    readln (st) ;
    for i := 1 to N do
        if st = a[i] then
            writeln (st, ' was fount at place: ', i) ;
:
End.

```

שאלה 5:

מ	i	פלט
0		3
1	1	
2	2	
3	3	

א. טבלת מעקב :

התכנית סופרת מדפיסה את מספר סימני ה-'x' באלכסון הראשי.

ב. אחרי השינוי יודפס מספר סימני ה-'x' בשורה 2 במטריצה.

פרק ב

שאלה 6:

{--- קלט גובה המבקרים בגן המשחקים ---}
 {--- קלט כמה גבוהים, כמה נמוכים ומהו הבונוס למפעילים ---}

```

program tar_6 ;
var
    height : real ;
    count_big, count_small : integer;
{-----}
{--- מסנתת קלט לבדיקת גובה לא שלילי ---}
function get_height : real ;
var
    height : real ;
begin
    repeat
        write ('Enter height ---> ');
        read (height) ;
    until height >= 0 ;
    get_height := height ;
end;
{-----}
Begin
    count_big := 0;
    count_small := 0;
    height := get_height ;
    while height <> 0 do
    
```

```

begin
  if height >= 1.70 then
    begin
      writeln ('big') ;
      inc (count_big);
    end
  else
    begin
      writeln ('small') ;
      inc (count_small) ;
    end ;
  height := get_height ;
end ;
writeln ('Number of BIG = ', count_big) ;
writeln ('Number of SMALL = ', count_small) ;
writeln ('Bonus for BIG = ', count_big div 15) ;
writeln ('Bonus for SMALL = ', count_small div 20) ;
End.
```

סעיף ב':

שאלה 7:

n	p	s	i	j	פלט	אלגוריתם א'
2	3	0			18	א. טבלת מעקב. ב. ההשמה התבצעה 6 פעמים.
		3	1	1		
		6		2		
		9		3		
		12	2	1		
		15		2		
		18		3		

n	p	s	k	פלט	אלגוריתם ב'
2	3	0		18	א. טבלת מעקב. ב. ההשמה התבצעה 3 פעמים. ג. אלגוריתם ב יותר יעיל כי הוא מבצע פחות איטרציות ("סיבובי לולאה").
		3	1		
		6	2		
		9	3		

$$n * p^2$$

האלגוריתם מחשב את

שאלה 8:

```

{--- הצפנת מחרוזת: ביטול תווי הרווח, ---}
{--- אחרי כל תו הגדול מ-M שותלים את התו הקודם לה ---}
{--- אחרי כל תו הקטן מ-M שותלים את התו העוקב לה ---}

program tar_8 ;
const N = 120 ;
var
    st : string [N] ;
    new_st : string [2*N] ;
    i, k : integer ;
Begin
    write ('Enter a string ---> ');
    readln (st) ;
    k := length (st) ;
    new_st := '' ;      { starting with an "empty" string }
    for i := 1 to k do
        if st[i] <> ' ' then    { if not a blank char }
            begin
                new_st := new_st + st[i] ;
                if st[i] < 'M' then
                    new_st := new_st + succ(st[i])
                else
                    new_st [j] := new_st + pred(st[i]);
            end ;
        writeln (new_st) ;
    End.

```

פרק ג**שאלה 9:**

מערך דו-ממדי המכיל מספרים שלמים : m (18×25)

- קלט למטריצה $m \leftarrow$
- מציאת איבר המינימום במטריצה \leftarrow לעמודה min_col , לשורה min_line
- הדפסת איברי המטריצה משורה min_line ועד שורה אחרונה ומעמודה min_col ועד עמודה אחרונה (מאיבר המינימום). עבור כל איבר יש לבדוק אם הוא זוגי. אם כל האיברים זוגיים יודפס "זוגי".

```
{--- קליטת מספרים למטריצה, מציאת המינימום ---}
{--- והדפסת הריבוע הנוצר מימין ומתחת לאיבר המינימום ---}
```

```
program tar_9 ;
const
    Line = 6 ;      { should be 18 }
    Col = 8 ;      { should be 25 }
type
    mat_type = array [1..Line, 1..Col] of integer ;
var
    m : mat_type ;
    i, j, min : integer;
    min_line, min_col : integer ;

{-----}
{--- filling matrix with number ---}
{--- במבחן לא היה צריך לקלוט המספרים למטריצה ---}

procedure mat_kelet (var m : mat_type) ;
var
    i, j : integer ;
begin
    for i := 1 to Line do
        for j := 1 to Col then
            begin
                write ('Enter a number → ');
                read (m[I,j]) ;
            end ;
        end ;
    end ;
end;
```

```

{-----}
{--- הפונקציה מקבלת מטריצה, מספר שורה ומספר עמודה ומדפיסה ---}
{--- איברי המטריצה מנקודה זו ועד פנה ימנית תחתונה של המטריצה ---}
procedure mat_pelet (m:mat_type; min_line, min_col:integer) ;
var   i, j : integer ;
      zugl : boolean ;
begin
  zugl := true ;
  for i := min_line to Line do
    begin
      for j := min_col to Col then
        begin
          write ( m[i,j] :5);
          if m[i,j] mod 2 = 1 then
            zugl := false ;           // נמצא איבר אי-זוגי
          end ;
        writeln ;
      end ;
    if zugl then
      writeln ('Zugi') ;
    end;
  {-----}
Begin
  mat_kelet (m) ;
  {--- looking for smallest element in matrix ---}
  min := m [1,1] ;
  min_line = 1;
  min_col = 1 ;
  for i := 1 to Line do
    for j := 1 to Col do
      if m[i,j] < min then
        begin
          min := m[i,j] ;
          min_line := i ;
          min_col := j ;
        end ;
      end ;
  mat_pelet (m, min_line, min_col) ;
End.

```

ניתן להפוך קטע זה לפרוצדורה המקבלת כפרמטר את המערך (לפי ערך) ואת משתני השורה ועמודה (לפי כתובת).
הפרוצדורה תשים במשתנים את מספר התא (שורה ועמודה) של הערך המינימלי.

שאלה 10:

טיפוס תלמיד (stud_type) = רשומה

	name	שם = מחרוזת.
	addr	כתובת = מחרוזת.
	id	מס' זיהוי = מספר ארוך.
{ 1 = שפת C, 2 = שפת פסקל }	prog_leng	שפת תכנות = מספר
	grad	ציון = מספר שלם.

stud = מערך בגודל 1750 שכל אחד מאבריו מטיפוס תלמיד.

- קלט נתוני התלמידים למערך stud
- חישוב הציון הממוצע \leftarrow avg
- פיצול התלמידים שקיבלו ציון גבוה מהממוצע לשני מערכים C, pas \leftarrow
- הדפסת מערך C (שבו j תלמידים)
- הדפסת מערך pas (שבו k תלמידים).

מערכים C ו-pas אף הם בגודל 1750 כל אחד.

```
{--- תחרויות ארציות במדעי המחשב: ---}
{--- קלט: פרטי התלמידים ---}
{--- פלט: רשימות הגבוהים מהממוצע לפי שפת תכנות ---}
```

```
program tar_10 ;
const N = 6 ;      { should be 1750 }

type
  stud_type = record
    name : string[30] ;
    addr : string[50] ;
    id : longint ;
    prog_leng : integer ;
    grad : integer ;
  end ;
  arr_type = array [1..N] of stud_type ;

var
  stud : arr_type ;
```

```

{-----}
{--- קלט פרטי התלמידים למערך ---}

procedure arr_kelet (var stud : arr_type ) ;
var
    T : stud_type ;
    i : integer ;
begin
    for i := 1 to N do
        begin
            write ('Name ---> ') ;
            readln (T.name) ;
            write ('Address ---> ') ;
            readln (T.addr) ;
            write ('ID num ---> ') ;
            readln (T.id) ;
            write ('Programming language (1 for C, 2 for Pascal) ---> ') ;
            readln (T.prog_leng) ;
            write ('grade ---> ') ;
            readln (T.grad) ;

            stud [i] := T ;
            writeln ;
        end ;
    end ;

{-----}
{--- הדפסת רשימת התלמידים במערך ---}
{--- קלט לפוקציה: מערך וגודלו ---}
{--- פלט: שם, כתובת ומספר זיהוי ---}

procedure arr_pelet (arr : arr_type ; size : integer) ;
var
    i : integer ;
begin
    for i := 1 to size do
        begin
            writeln ('name: ', arr[i].name) ;
            writeln ('address: ', arr[i].addr) ;
            writeln ('ID: %ld ', arr[i].id) ;
        end ;
    end ;
end ;

```

```

{-----}
{--- טענת כניסה: מערך פרטי התלמידים ---}
{--- טענת יציאה: מוחזר ממוצע הציונים ---}

function arr_avg_grad (stud : arr_type) : real ;
var
    i : integer ;
    sum : integer ;
begin
    sum := 0 ;
    for i := 1 to N do
        sum := sum + stud[i].grad ;
    arr_avg_grad := sum/N ;
end ;

{-----}
{----- מיון התלמידים שהצליחו למעוך פסקל ו-C ---}
{--- הפונקציה מקבלת מערך מאותחל של פרטי התלמידים ---}
{--- ממיינת את אילו שציוניהם מעל הממוצע למעוך פסקל ומעוך C ---}
{--- ומדפיסה אותם לפי שפה. ---}

procedure arr_select (stud : arr_type) ;
var
    pas, C : arr_type ;
    i, j, k : integer;
    avg : real ;
begin
    avg := arr_avg_grad (stud) ;
    j := 0 ;
    k := 0 ;
    for i := 1 to N do
        if stud[i].grad > avg then
            if stud[i].prog_leng = '1' then
                begin
                    inc (j) ;
                    C [j] := stud [i] ;
                end
            else
                begin
                    inc (k) ;
                    pas [k] := stud [i] ;
                end ;
            end ;
        end ;
    end ;
end ;

```

```
    arr_pelet (C, j) ;  
    arr_pelet (pas, k) ;  
end ;  
{-----}  
Begin  
    arr_kelet (stud) ;  
    arr_select (stud) ;  
End.
```