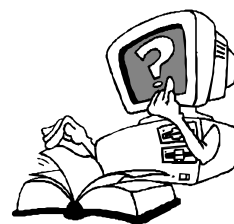


מועד הבחינה: 3 מאי, 2005



מדעי המחשב

2 יחידות לימוד

פרק ראשון - 50 נקודות

עליך לענות על כל השאלות 1 - 5 (לכל שאלה 10 נקודות)

שאלה 1:

לפניך האלגוריתם המילולי הבא, אשר הקלט שלו הוא שני מספרים שלמים חיוביים (כולל אפס):

- (1) הצב 1 במשתנה m .
- (2) קלוט שני מספרים שלמים למשתנים x ו- y .
- (3) הצב 1 במשתנה $count$.
- (4) כל עוד $(count \leq y)$ בצע:
 - (4.1) חשב את $m * x$ והצב במשתנה m .
 - (4.2) הגדל את ערכו של $count$ ב-1.
- (5) הצג כפלט את ערכו של m .

- א. עקוב בעזרת טבלת מעקב ורשום מה יהיה הפלט עבור הקלט $x = 2$ $y = 3$.
- ב. תן דוגמת קלט עבורה יוצג כפלט המספר 32.
- ג. תן דוגמת קלט עבורה גוף הלולאה לא יתבצע ורשום מה יוצג כפלט במקרה זה.

	m	x	y	count	$count \leq y$	פלט	
א. (האלגוריתם מחשב את x^y)	1	2	3	0	כן	8	
	2			1	כן		
	4			2	כן		
	8			3	לא		

ב. $x = 2$, $y = 5$

ג. לכל $y = 0$, $x \neq 0$ יוצג כפלט הערך 1

שאלה 2:

לפניך קטע תוכנית:

C:	פסקל:
<pre>scanf("%d %d %d", &a, &b, &c); if (c < b && c > a) printf(" * "); else if (c > b c < a) printf(" # ");</pre>	<pre>read (a, b, c); if (c < b) and (c > a) then writeln (' * '); else if (c > b) or (c < a) then writeln (' # ');</pre>

- א. תן דוגמה לערכי קלט עבור a, b, c עבורם תודפס כוכבית. (השורה המסומנת ב- {1})
 ב. תן דוגמה לערכי קלט עבור a, b, c עבורם תודפס סולמית. (השורה המסומנת ב- {2})
 ג. האם קיים קלט כלשהו עבורו לא מתבצע פלט כלל? אם כן - תן דוגמה מספרית, אם לא - הסבר מדוע.

<p>$a = 1, b = 5, c = 3$ בולטא:</p> <p>לכל c</p> <p>$a = 2, b = 6$ בולטא:</p> <p>כל c שנבחר השונה מ- a ומ- b</p> <p>$c > a$ וגם $c = b$ או $c < b$ וגם $c = a$</p>	<p>א. $a < c < b$</p> <p>ב. $c < a$ $b < c$</p> <p>$a = 2, b = 6$ בולטא:</p> <p>$c = 1$ או $c = 9$</p> <p>ג. כל שלושה מספרים a, b, c המקיימים: $c = a$ או $c = b$ או $c = a$</p>
--	---

שאלה 3

כתוב תוכנית מחשב שתקלוט 200 מספרים חיוביים. התכנית תדפיס כפלט כמה מהמספרים שנקלטו הם תלת-ספרתיים וגם מתחלקים ב-7 ללא שארית.

C:

```
// קלט: סדרה של 200 מספרים .  
// פלט: כמה מהמספרים הם תלת ספרתיים וגם מתחלקים ב-7 .  
  
#include <stdio.h>  
  
void main ()  
{  
    int num, i, count = 0;  
    for (i = 1 ; i <= 200 ; i++)  
    {  
        printf (" הקש מספר → ");  
        scanf ("%d", &num);  
        if (num > 99 && num < 1000 && num%7 == 0)  
            count ++;  
    }  
    printf (" מספרים תלת ספרתיים המתחלקים ב-7 %d \n", count);  
}
```

פסקל:

```
{ קלט: סדרה של 200 מספרים . }  
{ פלט: כמה מהמספרים הם תלת ספרתיים וגם מתחלקים ב-7 . }  
  
program numbers ;  
var  
    num , count , i : integer;  
  
begin  
    count := 0;  
    for i := 1 to 200 do  
        begin  
            write (' הקש מספר → ');  
            read (num);  
            if (num > 99) and (num < 1000) and (num mod 7 = 0) then  
                count := count + 1;  
        end;  
    writeln (' מספרים תלת ספרתיים המתחלקים ב-7 ', count);  
end.
```

שאלה 4

1. נתון מערך A בגודל 13 מכיל מספרים שלמים וחיובים:

	0	1	2	3	4	5	6	7	8	9	10	11	12
a	10	8	15	45	11	55	7	6	30	25	34	1	5

לפניך שלושה קטעים:

C	פסקל
I for (i = 0 ; i < 13 ; i++) if (a[i] % 5 == 0) printf ("%d", i);	I for i := 0 to 13 do if a[i] mod 5 = 0 then write (i);
II for (i = 0 ; i < 13 ; i++) if (i % 5 == 0) printf ("%d", a[i]);	II for i := 0 to 13 do if i mod 5 = 0 then write (a[i]);
III int i = 0; while (i < 13) { printf ("%d", a[i]); i = i+5; }	III i := 0; while (i < 13) do begin write (a[i]); i := i+5; end;

א. מה יודפס אחרי הרצת כל אחד משלוש הקטעים הני"ל?

- I ⇒ 0 3 5 8 9 12
- II ⇒ 10 55 34
- III ⇒ 10 55 34

ב. האם קיימים קטעים המשיגים את אותה מטרה? אם כן, מהם.

כן II ו-III

ג. משני הקטעים שציינת בסעיף ב', איזה קטע יעיל יותר. נמק.

קטע III יעיל יותר. קטע II עובר על כל איברי המערך. קטע III עובר רק על האיברים שבמקומות המתחלקים ב-5.

פסקל:

(* קלט: מחרוזת.
* פלט: האם במחרוזת מופיעה לפחות נקודה אחת, ולא בתחילת המחרוזת,
* ולא בספה ולא ברציפות.
*)

```
program sting_dot ;
var
  str : string [100];
  k, i : integer;
  valid, found : boolean;
begin
  valid := TRUE ;
  found := FALSE;
  write ('Type a string → ');
  readln (str);
  k := length(str);
  if (str[1] = '.') and (str[k] = '.') then valid := FALSE;
  i := 2;
  while (i < k and valid) do
    begin
      if (str[i] = '.')
        begin
          found := TRUE;
          if (str[i+1] = '.') then valid := FALSE;
        end;
      inc (i) ;
    end ;
  if (found and valid) then writeln ('Correct string ')
  else writeln ('Incorrect string ');
end.
```

קטע התכנית

פרק שני - 30 נקודות

ענה על שתיים מהשאלות 6 - 8 (לכל שאלה 15 נקודות)

שאלה 6:

הוחלט לארגן את מאגר שמות מדליקי המשואות ב-57 שנות המדינה במערך דו-ממדי מסדר 57×12 (57 שנים, 12 משואות בשנה לפי סדר הדלקתן).

- א. הגדר את מבנה הנתונים **מאגר שמות מדליקי המשואות** שבשאלה.
- ב. כתוב פונקציה שתקבל שם של מדליק משואה ואת מאגר השמות במאגר ותחזיר את השנה בה הוא הדליק משואה ואת מספר המשואה בסדר ההדלקה של אותה שנה. **הנחה**: כל משואה מודלקת על-ידי מדליק אחד בלבד.
- ג. כתוב תכנית (הכוללת בתוכה את סעיפים א ו-ב), הקולטת שם של מדליק משואה ומדפיסה את השנה בה הדליק את המשואה ומה היה מספר המשואה שהדליק. הנח שקיימת פונקציה הממלאת את מאגר השמות, ובתחילת התכנית המאגר מלא בנתונים.

C:

```
/* מטריצה של שמות מדליקי המשואות לפי שנים ולפי סדר ההדלקה  
* פלט: באיזו שנה ובאיזה סדר הדליקו משואה.  
*/
```

```
#include <string.h>  
#include <stdio.h>  
  
#define YEAR 57  
#define ORD 12  

```

--- קלט: שמות מדליקי המשואות לפי שנים ---

```
void mat_kelet (mat_type M)
```

```
{
    int i, j;
    for (i = 0 ; i < YEAR ; i++)
    {
        for (j = 0 ; j < ORD ; j++)
        {
            printf ("→ שם מדליק המשואה ?");
            gets (M[i][j]);
        }
    }
}
```

אין צורך לממש פונקציה זו
במבחן.

--- טענת כניסה: שם מדליק המשואה ---

--- טענת יציאה: השנה והסדר שבה הדליק את המשואה ---

```
int beacons_time (mat_type M, string_type name, int *yy)
```

```
{
    int i, j;
    int year, order;
    for (i = 0 ; i < YEAR ; i++)
    {
        for (j = 0 ; j < ORD ; j++)
            if (strcmp(name, M[i][j]) == 0)
            {
                year = i;
                order = j;
            }
    }
    *yy = year;
    return order;
}
```

פסקל:

(* קלט: מטריצה של שמות מדליקי המשואות לפי שנים ולפי סדר ההדלקה
* פלט: באיזו שנה ובאיזה סדר הדליקו משואה.
*)

```
program BEACONS ;
```

```
const
```

```
    YEAR = 57;  
    ORDER = 12;
```

```
type
```

```
    string [20] = string_type;  
    mat_type = array [1..YEAR, 1..ORDER] of string_type;
```

```
var
```

```
    M : mat_type;  
    name : string_type ;  
    yy, place : integer ;
```

{--- קלט: שמות מדליקי המשואות לפי שנים ---}

```
procedure mat_kelet (M : mat_type);
```

```
var
```

```
    i, j : integer;
```

```
begin
```

```
    for i := 1 to YEAR do  
        for j := 1 to ORDER do  
            begin  
                write ('→ שם מדליק המשואה ?');  
                readln (M[i][j]);  
            end;
```

```
end;
```

אין צורך לממש פונקציה זו
במבחן.

```
{--- טענת כניסה: שם מדליק המשואה ---}  
{--- טענת יציאה: השנה והסדר שבה הדליק את המשואה ---}  
procedure beacons_time (M : mat_type, name : string_type, var yy, place : integer) ;  
var  
    i, j : integer ;  
    for i := 1 to YEAR do  
        for j := 1 to ORDER do  
            if name = M[i][j] then  
                begin  
                    year := i;  
                    place := j;  
                end ;  
            end;  
        end;  
    end;  
  
begin  
    mat_kelet(M);  
    write ('שם מדליק המשואה?');  
    readln (name);  
  
    beacons_time (M, name, yy, place);  
    writeln (place, 'הדליק את המשואה ה-', name);  
    writeln (' בשנה ה-', yy, ' למדינה.');
```

שאלה 7:

נתון מערך A בגודל N (N אי זוגי) המכיל תווים, ונתונה הפונקציה סוד הבאה:

סוד(A)

{ טענת כניסה : השגרה מקבלת כפרמטר את מערך A. }

{ טענת יציאה : _____ }

(1) $k \leftarrow 1$

(2) $j \leftarrow 0$ { עבור פסקל: $j \leftarrow 1$ }

(3) כל עוד $k < N$, בצע:

(3.1) $k \leftarrow k + 2$

(3.2) $j \leftarrow j + 1$

(4) החזר $A[j]$.

א. מהו הערך שתחזיר הפונקציה עבור המערך A שלהלן שגודלו 9 איברים:

B Y N P R E D A H

ב. מהו הערך שתחזיר הפונקציה עבור המערך A שלהלן וגודלו 7.

e t u p m o c

ג. השלם את טענת היציאה של הפונקציה.

ד. קבע גודל מערך N ($N > 2$) ותן דוגמא למערך שעבורו יחזור הערך D.

ה. כתוב פונקציה יעילה יותר (ללא לולאה) להשגת אותה מטרה.

אינדקס המערך 0 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9

פסקל / C a B Y N P R E D A H

N	k	j	ערך מוחזר
9	1	0	A[4]
	3	1	מוחזר התו R
	5	2	
	7	3	
	9	4	

N	k	j	ערך מוחזר
9	1	1	A[4]
	3	2	מוחזר התו R
	5	3	
	7	4	
	9	5	

א.

ב. המערך השני יחזיר את התו 'ק'.

ג. טענת היציאה: מוחזר האיבר שנמצא בתא האמצעי המערך.

ד. כל מערך של תווים בגודל אי-זוגי שהאיבר האמצעי שלו הוא 'D' מתקבל.

ג. מכיוון שמדובר במערך של תווים ולא במחרוזת, מחזיר את התו שבמקום N חלקי 2.

טענת כניסה: פונקציה A בגודל N של תווים, N אי-זוגי.

טענת יציאה: מוחזר התו שבאמצע המערך.

C:	פסקל:
<pre>char middle_arr (arr_type A) { return A[N / 2]; }</pre>	<pre>function middle_arr (A : arr_type) : char; begin middle_arr := A[N div 2]; end;</pre>

שאלה 8:

מחרוזת תקרא "סימטרית חלקית" אם היא לא מחרוזת ריקה, מספר התווים בה הוא זוגי, האות הראשונה שווה לאות האחרונה, ושתי האותיות האמצעיות שוות זו לזו.

לדוגמא: המחרוזות הבאות: x1bdyyh5kx , DD , abba , abCCda
 הינן מחרוזות "סימטריות חלקיות".

כתוב תכנית הקולטת סדרה של מחרוזות, ומדפיסה את כל המחרוזות ה"סימטריות חלקיות" וכמה מחרוזות כאלו היו. סיום הקלט כאשר תיקלט המחרוזת "xxx".
 הנח שאורך כל המחרוזות קטן מ- 20 תווים.

C:

```
/* קלט: סדרה של מחרוזות.
 * פלט: כמה מהמחרוזות הינן "סימטריות חלקיות"
 * מחרוזת "סימטרית חלקית" הינה מחרוזת לא ריקה באורך זוגי
 * אשר התו הראשון בה שווה לאחרון ושני התווים האמצעיים שווים זה לזה.
 */
```

```
#include <stdio.h>
#include <string.h>

int main()
{
    int count = 0, mid, k;
    char str[20];

    printf("הקש מחרוזת → ");
    scanf("%s", str);

    while (strcmp(str, "xxx") != 0)
    {
        k = strlen(str);
        if (k != 0 && k%2 == 0)
        {
            mid = k/2;
            if ((str[0] == str[k-1]) && (str[mid-1] == str[mid]))
            {
                puts (str);
                count ++ ;
            }
        }
        printf("הקש מחרוזת → ");
        scanf("%s", str);
    }
    printf("מחרוזות סימטריות חלקיות %d היו \n", count);
}
```

פסקל:

```
(*
*                               קלט: סדרה של מחרוזות.
*                               פלט: כמה מהמחרוזות הינן "סימטריות חלקיות"
*                               מחרוזת "סימטרית חלקית" הינה מחרוזת לא ריקה באורך זוגי
*                               אשר התו הראשון בה שווה לאחרון ושני התווים האמצעיים שווים זה לזה.
*)

program SYMMETRIC_STRING ;
var
    count, mid, k : integer;
    str : string[20];
begin
    count := 0;
    write (' הקש מחרוזת → ');
    readln (str);
    while (str <> "xxx") do
        begin
            k := length (str);
            if (k <> 0) and (k mod 2 = 0) then
                begin
                    mid := k div 2;
                    if (str[1] = str[k]) and (str[mid] = str[mid+1]) then
                        begin
                            writeln (str);
                            inc (count) ;
                        end;
                    end;
                write (' הקש מחרוזת → ');
                readln (str);
            end;
        writeln (' מחרוזות סימטריות חלקיות ', count, ' היו ');
    end.
```

פרק שלישי - 20 נקודות

עליך לענות על אחת מהשאלות 9 - 10.

שאלה 9:

"מערך-מספרים-עוקבים" הוא מערך דו-ממדי של מספרים שלמים שבו בכל שורה המספרים עוקבים זה לזה.

לדוגמא:

5	6	7
21	22	23
-6	-5	-4

המערך הראשון הינו מערך-מספרים-עוקבים

1	2	4
43	44	45
6	7	8

ואילו המערך השני אינו מערך-מספרים-עוקבים

יש לכתוב תוכנית שתקלוט מספרים לתוך מערך דו-ממדי מסדר $N \times N$, תבדוק אם הוא מערך-מספרים-עוקבים ותודיע הודעה מתאימה.

- פרק את הבעיה לתת-בעיות, כך שכל תת-בעיה תיפתר באמצעות פונקציה. ציין את הכותרת של כל פונקציה, הפרמטרים שלה, וטענות כניסה ויציאה מפורטות.
- בחר משתנים עיקריים, הגדר את טיפוסיהם ותאר את תפקידיהם.
- כתוב תוכנית בשפת C לפתרון הבעיה.
 (אין צורך לחזור על טענות הכניסה והיציאה שפרטת בסעיף ב').



רצו חולביס!

פרוק לתת-בעיות:

- קלט: קלט למטריצה.
- עיבוד: בדיקת המטריצה. האם כל שורה מקיימת מספרים-עוקבים.
- פלט: הדפסת התוצאה.

- בשלב הראשוני נפרק כל בעיה לשלושה חלקים: קלט, עיבוד (חישוב) פלט.
- כל לולאה יכולה להפוך לתת-בעיה שהפתרון שלה יוצג באמצעות פונקציה או פרוצדורה.

קלט-למטריצה שם הפעולה: `mat_kelet`
 פעולה הממלאת מטריצה בגודל $N \times N$ במספרים שלמים.

שורת-מספרים-עוקבים (M, i) שם הפעולה: `line_sequential_num`
 טענת כניסה: מטריצה M ושורה i .
 טענת יציאה: מוחזר "אמת" אם כל איברי השורה עוקבים זה לזה, ו-"שקר" אחרת.

מטריצת-מספרים-עוקבים (M) שם הפעולה: `mat_sequential_num`
 טענת כניסה: מטריצה M בגודל $N \times N$.
 טענת יציאה: מוחזר "אמת" אם המטריצה היא **מטריצת-מספרים-עוקבים**, ו-"שקר" אחרת.

משתנים עיקריים:

שם המשתנה	טיפוס המשתנה	תפקיד המשתנה
M	מטריצה ריבועית מסדר $N \times N$	המטריצה הנבדקת.
i	int	מספר השורה הנבדקת
sequence	int	משתנה בוליאני המקבל ערך "אמת" אם יש מספרים עוקבים, ו-"שקר" אחרת.

C:

```

/* מטריצה מסדר  $N \times N$ 
 * פלט: הודעה מתאימה האם המטריצה היא מטריצת-מספרים-עוקבים
 */
#include <stdlib.h>
#include <stdio.h>

#define TRUE 1
#define FALSE 0

#define N 3
typedef int mat_type[N][N];

void mat_kelet (mat_type m);
int mat_sequence_num (mat_type m);
int line_sequence_num (mat_type m, int i);

int main()
{
    mat_type M;
    mat_kelet(M);
    if (mat_sequence_num(M) //--- if (mat_sequence_num(M) == TRUE)
        printf ("\n מערך מספרים עוקבים");
    else
        printf ("\n לא מערך מספרים עוקבים");
}
    
```

```
void mat_kelet (mat_type m)
{
    int i, j;
    for (i = 0 ; i < N ; i++)
        for (j = 0 ; j < N ; j++)
            {
                printf ("הקש מספר → ");
                scanf ("%d", &m[i][j]);
            }
}
```

```
int mat_sequence_num (mat_type m)
{
    int i, sequence = TRUE;
    for (i = 0 ; i < N ; i++)
        if (line_sequence_num(m,i) == FALSE) } (*)
            sequence = FALSE;
}
```

(*) צורה אלגנטית יותר לרישום הקטע המסומן:

```
for (i = 0 ; i < N ; i++)
    sequence = sequence && line_sequence_num(m,i);
```

```
int line_sequence_num (mat_type m, int i)
{
    int j, sequence = TRUE;
    for (j = 1 ; j < N ; j++)
        if (m[i][j] - m[i][j-1] != 1)
            sequence = FALSE;
    return sequence;
}
```

פסקל:

```
program Sequence_Matrix;
const N = 3;
type mat_type = array [1..N,1..N] of integer;
var M : mat_type;

procedure mat_kelet (var m : mat_type);
var i, j : integer;
begin
  for i := 1 to N do
    for j := 1 to N do
      begin
        write (' הקש מספר → ');
        read (m[i][j]);
      end;
    end;
  end;

function line_sequence_num (m : mat_type ; i : integer): boolean;
var j : integer;
    sequence : boolean ;
begin
  sequence := TRUE;
  for j := 1 to N do
    if (m[i][j] - m[i][j-1] <> 1) then
      sequence := FALSE;
  line_sequence_num := sequence;
end;
```

(*) צורה אלגנטית יותר לרישום הקטע המסומן:

```
for i := 0 to N do
  sequence := sequence and line_sequence_num(m,i);
```

```
function mat_sequence_num (m : mat_type): boolean;
var i : integer;
    sequence : boolean ;
begin
  sequence := TRUE;
  for i := 1 to N do
    if (line_sequence_num(m,i) = FALSE) then
      sequence := FALSE;
  mat_sequence_num := sequence;
end;
```

```
begin
  mat_kelet(M);
  if mat_sequence_num(M) then {--- if (mat_sequence_num(M) = TRUE) then ---}
    writeln (' מערך מספרים עוקבים ')
  else
    writeln (' לא מערך מספרים עוקבים ');
end.
```

שאלה 10:

במפעל מסויים יש 200 עובדים. שכרו של כל עובד מחושב כך: עבור כל אחת מ-160 שעות העבודה הראשונות, מקבל העובד 50 ₪ לשעת עבודה. עבור כל שעה מעבר לזה מקבל העובד 70 ₪ לשעת עבודה. לדוגמא:

עובד שעבר 184 שעות בחודש מסויים יקבל משכורת של 9680 ₪
המורכבת מ-: 160 ש"ע * 50 ₪ + 24 ש"ע * 70 ₪.

עליך לכתוב תכנית הקולטת עבור כל אחד מהעובדים את פרטי עבודתו בכל אחד מ-25 ימי העבודה בחודש מסויים. התכנית תקלוט בכל יום את שעת תחילת העבודה ואת שעת הסיום של כל עובד, ותחשב את מספר שעות העבודה החודשיות שלו.

הנח שהשעות מיוצגות ב-4 ספרות:

לדוגמא: עובד שהחל עבודתו בשעה 8 בבוקר וסיים בשעה 17, עבד בסה"כ 9 שעות באותו היום.

התכנית תציג כפלט את משכורתו של כל אחד מהעובדים ואת השכר הכולל ששילם המפעל בחודש זה.

א. פרק את הבעיה לתת-בעיות, כך שכל תת-בעיה תיפתר באמצעות פונקציה. ציין את הכותרת של כל פונקציה, הפרמטרים שלה, וטענות כניסה ויציאה מפורטות.

ב. בחר משתנים עיקריים, הגדר את טיפוסיהם ותאר את תפקידיהם.

ג. כתוב תוכנית בשפת C לפתרון הבעיה, תוך שימוש בתת הבעיות שהגדרת בסעיף א.



רצו חולכיס!

פרוק לתת-בעיות:

מבנה נתונים: מערך של צוברים. בכל תא במערך יישמרו שעות העבודה החודשיות של העובד.

קלט: קלט שעות העבודה של העובדים בחודש זה.

עבור כל יום מ-25 הימים, יש לעבור על כל 200 העובדים ולקבל את מספר שעות העבודה שלו ביום זה. מספר זה יתוסף למה שכבר נשמר במערך עד כה.

פעולה: מספר שעות העבודה של העובד ביום מסויים.

פעולה: מילוי מערך הצוברים בשעות העבודה של כל העובדים בחודש זה.

עיבוד: חישוב השכר לעובד

פלט: הדפסת המשכורות והסה"כ.

fill_working_hours :	שם הפעולה :	קלט-שעות-עבודה-חודשיות (A)
		מערך של צוברים A מאותחל לאפס.
		טענת כניסה :
		טענת יציאה :
		בתא i יהיו סך השעות שעבד עובד i בחודש מסויים.
get_hours :	שם הפעולה :	מספר-שעות
		מספר שעות העבודה של עובד ביום מסויים.
		טענת יציאה :
get_worker_salary :	שם הפעולה :	שכר-לעובד (num)
		מספר שעות העבודה של עובד בחודש מסויים.
		טענת כניסה :
		טענת יציאה :
		השכר המגיע לעובד עבור שעות עבודתו. (השכר במספרים שלמים).
arr_init :	שם הפעולה :	למתכנתי פסקל:
		אתחול-מערך
		כל תאי מערך A מכילים אפס.
		טענת יציאה :

C:

```

/*
 * קלט : שעות העבודה של 200 עובדי מפעל במשך 25 ימי עבודה.
 * פלט : השכר החודשי של כל עובד והשכר הכללי ששילם המפעל.
 */
#include <stdio.h>
#include <stdlib.h>

#define N 200
#define DAYS 25

typedef int arr_type [N];

int fill_working_hours (arr_type A);
int get_worker_salary (int num);
int get_hours ();

int main()
{
    int A[200] = {0};
    int i, j;
    int salary, total = 0;
    fill_working_hours (A);
    for (i = 0 ; i < N ; i++)
    {
        salary = get_worker_salary(A[i]);
        printf("%d : היא %d משכורתו של עובד", salary, i);
        total = total + salary;
    }
    printf ("%d : שילם המפעל", total);
}
    
```

```
void fill_working_hours (arr_type A)
{
    int i, j;
    int work_hour ;
    for (i=1 ; i <= DAYS ; i++)
    {
        for (j = 0 ; j < N ; j++)
        {
            work_hour = get_hour ();
            A[j] = A[j] + work_hour;
        }
    }
}

int get_hours ()
{
    int start_hour, fin_hour;
    printf ("הקש שעת תחילת העבודה → ");
    scanf ("%d", &start_hour);
    printf ("הקש שעת סיום העבודה → ");
    scanf ("%d", &fin_hour);
    return (fin_hour - start_hour);
}

int get_worker_salary (int num)
{
    int salary, over_time;
    if (num > 160)
    {
        over_time = num - 160 ;
        salary = 160*50 + over_time*70;
    }
    else
        salary = num*50;
    return salary;
}
```

פסקל:

(* קלט : שעות העבודה של 200 עובדי מפעל במשך 25 ימי עבודה.
* פלט : השכר החודשי של כל עובד והשכר הכללי ששילם המפעל.
*)

```
program workers ;
const
    days = 25;
    N = 200;

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

var
    A : arr_type;
    i, total, salary : integer;

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

function get_hours : integer;
var
    strt, fin : integer ;
begin
    write (' הקש שעת תחילת העבודה ');
    read (strt);
    write (' הקש שעת סיום העבודה ');
    read (fin);

    get_hours := fin - strt;
end;

procedure fill_working_hours (var a : arr_type);
var
    i, j : integer ;
begin
    arr_init (a);

    for i := 1 to days do
        for j := 1 to N do
            a[j] := a[j] + get_hours();
end;
```

```
function get_worker_salary (num : integer):integer;
var
    sum : integer;
    salary : integer;
begin
    if num > 160 then
        salary := 160 * 50 + (num-160) * 70
    else
        salary := num * 50;
    get_worker_salary := salary;
end;

begin
    fill_working_hours (a);
    total := 0 ;
    for i := 1 to N do
        begin
            salary := get_worker_salary (A[i]);
            writeln (salary, ' : היא ', i, ' מספר של עובד');
            total := total + salary;
        end;
    writeln (total, ' : סך השכר ששילם המפעל בחודש זה הוא:');
end.
```