

באיור לשאלה 5 נתון מערך חד־ממדי הכולל 16 איברים מטיפוס שלם (int).
נגדיר בשאלה הזו "זוג איברים צמודים" כזוג איברים עוקבים שמיקומו של הראשון מביניהם הוא זוגי. למשל: זוג האיברים במקומות ה־[2] וה־[3].



איור לשאלה 5

כתוב תכנית בשפת C, שתבצע את הפעולות האלה:

1. תגדיר מערך חד־ממדי הכולל 16 איברים מטיפוס שלם.
2. תקלוט מהמקלדת 16 נתונים ותציב אותם בתוך תאי המערך בזה אחר זה.
3. תחשב כמה זוגות של איברים צמודים שסכום התכנים שלהם שווה ל־2014 יש במערך.
4. תציג על צג המחשב:

I. את איברי המערך ואת תכניהם, כל איבר בשורה נפרדת, באופן הבא:

```
arr[0]=.....  
arr[1]=.....  
:  
:  
arr[15]=.....
```

II. את כמות (מספר) זוגות האיברים הצמודים, שסכום תכניהם הוא 2014, באופן הבא:

Number of couples of cells equal to 2014 =

שאלה 5

תשובה 5

1. int arr[16];

2. בהנחה שהוגדרו קבצי הכותר המתאימים:

```
int x,counter;  
for (x=0;x<16;x++)  
{  
    printf("\nPlease Enter a number for arr[%d] : ");  
    scanf("%d",&arr[i]);  
}
```

.3

```
for(x=counter=0;x<16;x=x+2) // מקדמים ב 2 כדי להגיע לזוג האברים הבא
    if(arr[x]+arr[x+1] == 2014)
        counter++;
```

.4 .א

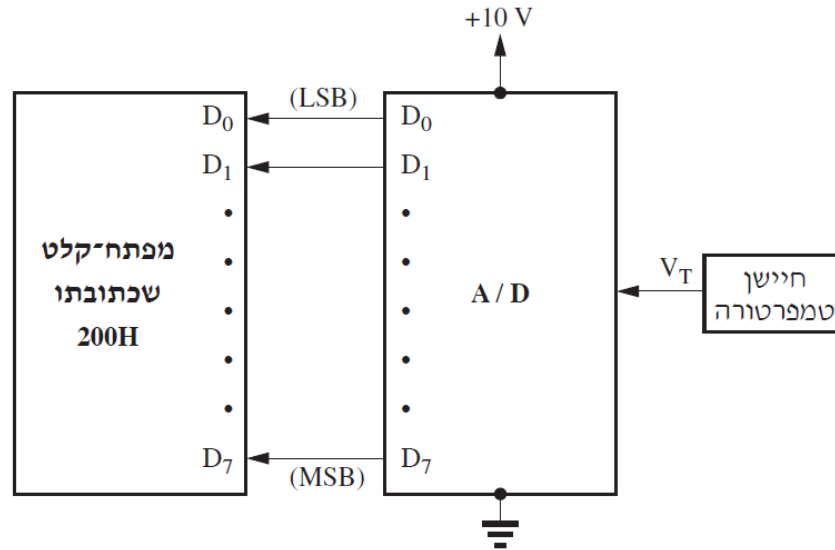
```
for(x=0;x<16;x++)
    printf("\narr [%d] = %d", x,arr[x]);
```

.ב

```
printf("Number of couples of cells equal to 2014 = %d", counter);
```

שאלה 6

באיור לשאלה 6 מתואר חיבור עקרוני בין מפתח־קלט שכתובתו 200H ובין ממיר אות תקבילי לאות ספרתי (A/D). במבוא הממיר מתקבל מתח V_T מחיישן טמפרטורה.



איור לשאלה 6

כאשר המתח במבוא הממיר הוא $V_T = 0\text{ V}$, הנתון שיפיק הממיר במוצאו יהיה 0.
 כאשר המתח במבוא הממיר הוא $V_T = 5.1\text{ V}$, הנתון שיפיק הממיר במוצאו יהיה $(255)_{10}$.

כאשר הטמפרטורה היא 0°C – המתח V_T יהיה 0 V.
 המתח V_T עולה ב- 0.1 V כאשר הטמפרטורה עולה ב- 1°C .

כתוב תכנית בשפת C, שתבצע את הפעולות האלה:

1. תציג על צג המחשב את הטמפרטורה המרבית (ב- $^\circ\text{C}$) שממיר ה-A/D יכול לקלוט.
2. תקלוט, בזו אחר זו, עשר דגימות של הנתון במוצא הממיר. לאחר כל פעולת קלט מהממיר, התכנית תמתין חמש שניות.
3. תמצא את הנתון הגדול ביותר שנדגם.
4. תחשב את הטמפרטורה (ב- $^\circ\text{C}$) המתאימה לנתון הגדול ביותר שנדגם, ותציג אותה על צג המחשב.

תשובה 6

מהנתונים של הממיר נחשב את הרזולוציה : $5.1 / 256 = 0.0199217875$
נניח שהוגדרו קבצי הכותר המתאימים.

1. את הטמפרטורה המרבית שניתן למדוד נחשב לפי המתח המקסימאלי שניתן למדוד עם הממיר שהוא 5.1V (לחלק במקדם הטמפרטורה של החיישן $(0.1\text{V}/^\circ\text{C})$: $5.1 / 0.1 = 51^\circ\text{C}$

מכאן שההדפסה תהיה :

```
printf("\nThe MAXIMUM temperature we can get is %f °c ", 5.1/0.1);
```

```
int adc[10],max,x;
for (x=0 ; x < 10 ; x++ )
{
    adc[x]=Inp32(0x200);
    sleep(5000);
}
for(max=adc[0],x=1;x<10; x++)
    if(max<adc[x])
        max=adc[x];
```

.4

אם נכפיל את המספר המקסימאלי שקיבלנו מהממיר (נמצא במשתנה max) ברזולוציה נקבל את המתח האנלוגי המתאים לו שהחיישן טמפרטורה הוציא. אם נחלק מספר זה במקדם הטמפרטורה של החיישן נוכל לדעת את הטמפרטורה.

```
printf( "\n The maximum temperature measured is : %f " , max * (5.1 / 256) / 0.1 );
```

```
1. #include <stdio.h>
2. #include <windows.h>
3. short_stdcall Inp32(short PortAddress);
4. void_stdcall Out32(short PortAddress,short data);
5. int read_Switch()
6.     {
7.     int sw,i,test=1,count=0;
8.     sw = Inp32(0x300);
9.     for(i=0;i<8;i++)
10.        {
11.        if((sw & test)==0)
12.            {
13.            count++;
14.            }
15.        test=test<<1;
16.        }
17.     return count;
18.     }
19. void main(void)
20. {
21.     int i, num, led=1;
22.     num=read_Switch();
23.     for(i=0;i<num ;i++)
24.     {
25.     Out32(0x302,led);
26.     Sleep(1000);
27.     led=(led*2)+1;
28.     }
29. }
```

א. הסבר את ההוראות שבשורות 8, 11, 15, 25 ו-27.

ב. הסבר מה מבצעת הפונקציה read_Switch().

ג. הסבר מה מבצעת התכנית כולה.

ד. סוגרים את המתגים S_1 , S_2 ו- S_3 . אילו נוריות LED יידלקו ובמשך כמה זמן תדלוק כל אחת מהן ב-3 השניות שלאחר סגירת המתגים הללו?

תשובה 7

א.

שורה 8

```
sw = Inp32(0x300);
```

זימון / קריאה לפונקציה Inp32. שולחים לפונקציה את הנתון 300H. הפונקציה יודעת להכניס נתון (מטיפוס short) מפורט קלט הנמצא בכתובת שהיא קיבלה 300H. את הנתון הזה הפונקציה מחזירה למשתנה sw.

שורה 11

```
if ( ( sw & test ) == 0)
```

בצע פעולת AND בין הביטים המתאימים במשתנה sw עם הביטים המתאימים במשתנה test (D0 במשתנה sw עם D0 במשתנה test, D1 עם D1 וכך הלאה). אם תוצאת פעולת ה AND היא 0 עבור לבצע את השורות 12 עד 14. אם התוצאה איננה 0 עבור לשורה 15.

שורה 15

```
test = test <<1 ;
```

בצע הזזה שמאלית פעם אחת לערך שיש במשתנה test ואת תוצאת ההזזה העבר ל test. ההזזה השמאלית מתבצעת על ידי העברת כל ביט פעם אחת ימינה. ביט ה MSB "נופל" וביט ה LSB מקבל 0. למעשה פעולה זו מכפילה את הערך שיש במשתנה test ב 2.

שורה 25

```
Out32 ( 0x302, led ) ;
```

קריאה/זימון לפונקציה Out32. לפונקציה שולחים 2 ערכים. הערך הראשון הוא 302H הערך השני הוא הערך הנמצא במשתנה led. הפונקציה מוציאה את הערך השני – הערך שב led - אל התקן פלט שכתובתו 302H.

שורה 27

```
led = ( led * 2 ) +1 ;
```

הכפל ב 2 את הערך שיש במשתנה led והוסף לתוצאה 1. את התוצאה העבר למשתנה led.

ב.

הפונקציה read_Switch מבצעת את הפעולות הבאות :

א. מגדירה 4 משתנים מטיפוס שלם. ב. מזמנת/קוראת לפונקציה Inp32 המכניסה אל המשתנה sw את מצב 8 המפסקים שבפורט הקלט שנמצא בכתובת 300H במרחב הקלט/פלט. ג. מבצעת בלולאת for, 8 פעמים, פעולת AND ביטים בין הנתון במשתנה test והנתון במשתנה sw. בכל פעם היא מזיזה את הנתון שבמשתנה test שמאלה פעם אחת. בכל פעולת AND כזו היא בודקת האם התוצאה 0 ואם כן היא מגדילה את הערך במשתנה count (הסבר מפורט בהמשך הסעיף). ד. מחזירה את הערך שבמשתנה count אל השורה הקוראת (שורה 22 בתוכנית. הערך שמוחזר נכנס אל המשתנה num). הסבר לולאת ה for ופעולת ה AND .

נניח שהמשתנה sw קיבל את מצב המפסקים וכרגע המצב בו הוא :

הערה : נתייחס רק ל 8 הביטים הנמוכים של המשתנה ונתעלם מהביטים הגבוהים יותר כי במערכת ההפעלה DOS משתנה מטיפוס שלם - int - "תופס" 2 בתים בזיכרון ובמערכת ההפעלה windows הוא "תופס" 4 בתים.

כאשר $i=0$ ונכניס את מצב המפסקים נקבל :

$$\begin{array}{r} \text{sw} = \text{S7 S6 S5 S4 S3 S2 S1 S0} \\ \underline{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1} \\ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \text{S0} \end{array} \quad \begin{array}{l} \text{היות ובמשתנה count יש בהתחלה 1 אז פעולת ה AND} \\ \text{תיתן תוצאה של :} \end{array}$$

אם S0 סגור (נותן 0 בכניסת פורט הקלט) , נקבל תוצאה 0 ואז נגדיל את המשתנה count ב 1 .
כרגע יהיה בו 1 . אם S0 פתוח לא נגדיל את ערך המשתנה count ויישאר בו 0 .
נזיז שמאלה פעם אחת את הערך שיש במשתנה test (הכפלה פי 2) וערכו יהיה 2 . שוב נבצע פעולת AND :

$$\begin{array}{r} \text{sw} = \text{S7 S6 S5 S4 S3 S2 S1 S0} \\ \underline{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0} \\ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \text{S1} \ 0 \end{array} \quad \begin{array}{l} \text{היות ובמשתנה count יש עכשיו 2 אז פעולת ה AND} \\ \text{תיתן תוצאה של :} \end{array}$$

אם S1 סגור (נותן 0 בכניסת פורט הקלט) , נקבל תוצאה 0 ואז נגדיל את המשתנה count ב 1 .
אם S1 פתוח לא נגדיל את ערך המשתנה count . שוב נזיז את המשתנה test ונחזור על השלבים
הקודמים עד שהמשתנה $i=8$ ואז מסתיימת לולאת ה for ומחזירים את ערך count לשורה הקוראת.
מכאן שהפונקציה בודקת כמה מפסקים סגורים (מכניסים 0).

ג.

1. התוכנית מתחילה בפונקציה main() שמגדירה משתנים וקוראת לפונקציה read_Switch .
2. פונקציה זו מכניסה את מצב המפסקים וסופרת את כמות המפסקים הסגורים (המפסקים שנותנים 0 בכניסת פורט הקלט) ומחזירה את המספר לפונקציה הראשית למשתנה num .
3. לאחר מכן מתבצעת לולאת for שבה מדליקים את הledים בצורה טורית לפי הקיים ב num . אם המספר ב num הוא 3 אז בהתחלה מדליקים את הled הראשונה (LED0) לשנייה אחת, אח"כ מדליקים 2 לדים - LED0 ו LED1 לשנייה נוספת ואח"כ נדלקות 3 לדים LED0 LED1 LED2 ונשארות דלוקות.

ד.

היות וסגרנו 3 מפסקים אז צורת ההדלקה תהיה כמו שהוסבר בסיום הסעיף הקודם.

להלן תכנית הכתובה בשפת C :

```
1.  #include <stdio.h>
2.  int convert(int *p_arr,int x)
3.  {
4.  int z=0;
5.  while (x>0)
6.      {
7.      *(p_arr+z)=x%2;
8.      x=x/2;
9.      z++;
10.     }
11. return z;
12. }
13. void main(void)
14. {
15. int i,num,w;
16. int arr[16]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
17. do
18.     {
19.     printf("ENTER AN INTEGER POSITIVE VALUE: \n");
20.     scanf("%d",&num);
21.     }
22. while(num<0);
23. w=convert(arr,num);
24. for (i=0;i<w;i++)
25.     {
26.     printf("%d",arr[w-1-i]);
27.     }
28. }
```

א. הסבר את ההוראות שבשורות 2, 7, 8, 20 ו-23.

ב. המשתמש הקליד את הנתון 46. מה יהיה פלט התכנית?

הערות: – השתמש בטבלת-מעקב עבור המשתנים x, z ואיברי המערך arr.

– שים לב שהמשתנה x הוא מטיפוס שלם (int).

ג. הסבר מה מבצעת הפונקציה convert(). נמק את תשובתך.

ד. הסבר מה מבצעת התכנית כולה.

תשובה 8

א.

שורה 2

```
int convert(int *p_arr, int x)
```

הגדרה של פונקציה בשם convert . הפונקציה מחזירה ערך מטיפוס שלם - int . היא מקבלת 2 ערכים. ערך אחד הוא כתובת הנכנסת למשתנה שהוא מצביע בשם p_arr מטיפוס שלם והערך השני הוא ערך הנכנס למשתנה x שהוא מטיפוס שלם .

שורה 7

```
*(p_arr+z) = x % 2 ;
```

פעולת שארית - מודולו - האומרת בצע חלוקה ב 2 של הערך שבמשתנה x ואת השארית (שהיא הערך בביט ה LSB של x) העבר לכתובת שעליה מצביעים הערך שבמצביע p_arr ועוד הערך שבמשתנה z . החיבור מבין המצביע לערך שב z מתבצע ביחידות לוגיות. אם z=1 אז מתחבר לערך שב p_arr ארבע כי ב windows משתנה מטיפוס int "תופס" 4 כתובות בזיכרון (ב DOS מתווסף רק 2 !) .

שורה 8

```
x = x / 2 ;
```

חלק ב 2 את הערך שב x ואת תוצאת החלוקה העבר ל x .

שורה 20

```
scanf("%d",&num);
```

זימון/קריאה לפונקציה scanf() . הפונקציה יודעת לקלוט מהמשתמש ערך עשרוני (לפי %d) ולהכניס ערך זה לכתובת (לפי האמפר-סנט &) של המשתנה num . כלומר אחרי החזרה מהפונקציה יהיה במשתנה num את הערך העשרוני שהמשתמש הקיש .

שורה 23

```
w=convert((arr,num);
```

קריאה/זימון לפונקציה convert . שולחים אליה 2 ערכים . הראשון הוא שם המערך arr (שם של מערך הוא הכתובת של האיבר הראשון במערך!) , כלומר את הכתובת של האיבר הראשון של המערך. כמו כן שולחים אליה את הערך שנמצא במשתנה num . הערך המוחזר מהפונקציה נכנס למשתנה w .

סעיפים ב + ג.

הערך 46 נשלח לפונקציה convert . ניעזר בטבלה הבאה המתארת משמאל לימין את השינויים הקורים בכל פעם שהתוכנית עוברת על השורות 5 עד 10 (כל עוד $x > 0$):

מצב התחלתי	מעבר 1	מעבר 2	מעבר 3	מעבר 4	מעבר 5	מעבר 6
$x \% 2$	$46 \% 2 = 0$	$23 \% 2 = 1$	$11 \% 2 = 1$	$5 \% 2 = 1$	$2 \% 2 = 0$	$1 \% 2 = 1$
$x = 46$	$46 / 2 = 23$	$23 / 2 = 11$	$11 / 2 = 5$	$5 / 2 = 2$	$2 / 2 = 1$	$1 / 2 = 0$
$z = 0$	1	2	3	4	5	6
$arr[0] = 0$	0					
$arr[1] = 0$		1				
$arr[2] = 0$			1			
$arr[3] = 0$				1		
$arr[4] = 0$					0	
$arr[5] = 0$						1
$arr[6] = 0$						
$arr[7] = 0$						

⋮						
⋮						
arr[15] = 0						

בתוכנית רואים שמחלקים את המספר ב 2 ואת השארית מכניסים למערך. שוב מחלקים ב 2 ואת השארית לאיבר הבא במערך וכך הלאה כל עוד $x > 0$.

בטבלה רואים את המעברים המתארים את השינוי במשתנים ובמערך. רק השינויים נראים בטבלה!. כאמור – כדי להמיר – convert – מספר עשרוני לבינארי מחלקים אותו ב 2 ורושמים את השארית. יורדים לשורה הבאה, מחלקים שוב ב 2 ושוב רושמים את השארית. כך עושים עד שהמספר שווה ל 0. את המספר הבינארי המתקבל קוראים מהשורה האחרונה כלפי מעלה.

המספר 46 בבינארי הוא : 101110 . אם נתבונן במערך מאיבר arr[5] עד arr[0] נראה שקיבלנו את המספר העשרוני 46.

בלולאת ההדפסה רושמים את הספרות מ arr[5] ועד arr[0] ולכן במסך המחשב נקבל : 101110

ד.

התוכנית הראשית מגדירה מספר משתנים ומערך בן 16 איברים כולם מטיפוס שלם. היא מבקשת מהמשתמש להקיש מספר חיובי שלם. התוכנית בודקת שהמשתמש הקליד מספר גדול מ 0. אם לא היא מבקשת ממנו שוב להקיש מספר חיובי שלם. את המספר היא מכניסה למשתנה num.

אם המשתמש הקיש מספר שלם היא קוראת לפונקציה convert, שולחת לה את המספר שהמשתמש הקיש ואת הכתובת של המערך. הפונקציה מפרקת את המספר לספרות הבינאריות שלו ומכניסה אותן בהתאמה, אחת אחרי השנייה לאיברי המערך. בסיום הפונקציה היא מחזירה ערך (מתואר בטבלה על ידי z) המראה לכמה איברים במערך הוכנסו הספרות. ערך זה נכנס למשתנה w. לאחר מכן מתבצעת לולאת הדפסה המדפיסה את איברי המערך המכילים את המספר הבינארי החל מהאיבר הראשון שיש בו ערך של '1', כי אין טעם לרשום את כל האיברים המכילים 0 משמאל לספרה הראשונה.