

מעריך חד-ממדי

תרגילים ופתרונות

חלק א' – מבוא למעריך חד-ממדי

- צפה בסרטון [מבוא למעריך חד-ממדי](#) (7:30 דקות).
- צפה בסרטון [המשך למעריך חד-ממדי](#) (5:30 דקות).

ענה על השאלות הבאות :

1. מה לדעתך, היתרון בשימוש במעריך ?
או במילים אחרות, באילו מקרים כדאי להשתמש במעריך ?
2. מה מייצג המושג element ? מהו אינדקס של element ? וכיצד הוא נספר במעריך ?
3. באילו מהמקרים הבאים כדאי לעבוד עם מעריך ? נמק.
- 3.1. ציונים של כיתה בת 20 סטודנטים.
- 3.2. פרטים של סטודנט (שנת לידה, ממוצע, מספר טלפון, גובה, משקל).
- 3.3. טמפרטורה של כל יום בחודש אוגוסט.
- 3.4. מספר שעות העבודה שעבדתי השבוע ושכר לשעה.
4. צייר את המעריך המתקבל עקב כל אחת מההגדרות הבאות :

4.1. `int ary[5] ;`

4.2. `int vec[10] = {3,-8, 55} ;`

4.3. `float m[4] = {1.5 , 7.9 , -0.8 , 3} ;`

4.4. `int aaa[200] = {0} ;`

4.5. `int s[5] , x ;`
`for (x=0 ; x<5 ; x++)`
`s[x] = x*2 ;`

4.6. `int s[5] , x ;`
`for (x=4 ; x>=0 ; x--)`
`s[x] = x+1 ;`



פתרון לחלק א'

1. מה לדעתך, היתרון בשימוש במערך ?
או במילים אחרות, באילו מקרים כדאי להשתמש במערך ?
תשובה : מואף להשתמש במערך כאשר ברצוננו לטפל בכמות גדולה של נתונים מאותו טיפוס ובעלי מנה (תפקיד) משותף.
2. מה מייצג המושג element ? מהו אינדקס של element ? וכיצד הוא נספר במערך ?
תשובה : מושג זה מתייחס לאיבר, תא בודד במערך.
אינדקס הוא המיקום של האיבר במערך, כאשר האיבר הראשון מיקומו 0, האיבר השני מיקומו 1 וכן הלאה.
3. באילו מהמקרים הבאים כדאי לעבוד עם מערך ? נמק.
 - 3.1. ציונים של כיתה בת 20 סטודנטים.
תשובה : כדאי, כי מדובר בכמות גדולה של נתונים הומולוגיים.
 - 3.2. פרטים של סטודנט (שנת לידה, ממוצע, מספר טלפון, גובה, משקל).
תשובה : לא כדאי, כי מדובר בנתונים שאינם הומולוגיים.
 - 3.3. טמפרטורה של כל יום בחודש אוגוסט.
תשובה : כדאי, כי מדובר בכמות גדולה של נתונים הומולוגיים.
 - 3.4. מספר שעות העבודה שעבדתי השבוע ושכר לשעה.
תשובה : אין צורך, מדובר במצב נתונים ואסוציאטיבי.

4. צייר את המערך המתקבל עקב כל אחת מההגדרות הבאות :

4.1. `int ary[5] ;`

0	1	2	3	4
?	?	?	?	?

4.2. `int vec[10] = {3,-8, 55} ;`

0	1	2	3	4	5	6	7	8	9
3	-8	55	0	0	0	0	0	0	0

4.3. `float m[4] = {1.5 , 7.9 , -0.8 , 3} ;`

0	1	2	3
1.5	7.9	-0.8	3.0

4.4. `int aaa[200] = {0} ;`

0	1	2	199
0	0	0	0

4.5. `int s[5] , x ;`

`for (x=0 ; x<5 ; x++)`

`s[x] = x*2 ;`

0	1	2	3	4
0	2	4	6	8

4.6. `int s[5] , x ;`

`for (x=4 ; x>=0 ; x--)`

`s[x] = x+1 ;`

0	1	2	3	4
1	2	3	4	5



חלק ב' – תרגול

שאלה 1

כתוב תוכנית המבצעת את המשימות הבאות :
א. קולטת ערכים לשני מערכים, בני 10 תאים מטיפוס שלם, כל אחד.
ב. מחברת את שני המערכים לתוך מערך שלישי, בהתאמה (ראה דוגמא).
ג. מדפיסה את המערך השלישי למסך.

לדוגמא :

vec1	1	6	-3		7
	vec1[0]	vec1[1]	vec1[2]		vec1[9]

Vec2	25	-4	10		0
	vec2[0]	vec2[1]	vec2[2]		vec2[9]

לאחר החיבור :

Vec3	26	2	7		7
	vec3[0]	vec3[1]	vec3[2]		vec3[9]

שאלה 2

הוסף לתכנית משאלה 1 קטע קוד הסורק את המערך השלישי `vec3[]` ומאתר את הערך הזוגי הגדול ביותר :
אם נמצא – יודפס המיקום שלו.
אם לא נמצא ערך זוגי במערך – יודפס -1.

לדוגמא, עבור המערך הבא יודפס 6, כי הערך 32 הוא הזוגי הגדול ביותר :

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
vec3	4	9	1	45	7	11	32	6	15	2

לדוגמא, עבור המערך הבא יודפס -1, כי לא קיים ערך זוגי במערך :

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
vec3	23	9	1	45	7	11	35	3	15	21



שאלה 3

הוסף לתכנית משאלה 1 קטע קוד הסורק את המערך השלישי `vec3[]`, בודק כל זוג שכנים במערך ומונה כמה איברים מהווים את השורש הריבועי של השכן שלהם.

לדוגמא, עבור המערך הבא יודפס 3 :

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
vec3	49	7	10	15	2	4	100	11	36	6

נימוק : קיימים 3 זוגות שכנים שאחד מהם מהווה שורש ריבועי של שכנו :
 $v[1]=7$ הוא שורש ריבועי של $v[0]=49$.
 $v[4]=2$ הוא שורש ריבועי של $v[5]=4$.
 $v[9]=6$ הוא שורש ריבועי של $v[8]=36$.

שאלה 4

הוסף לתכנית משאלה 1 קטע קוד הסורק את המערך השלישי `vec3[]` והופך את סדר הערכים.

לדוגמא, אם המערך הכיל את הערכים :

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
vec3	3	7	12	-4	5	-17	0	38	-21	22

בסיום התכנית הוא יכיל :

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
vec3	22	-21	38	0	-17	5	-4	12	7	3

פתרון לחלק ב'

```
#include <stdio.h>
#define N 4
int main()
{
    int vec1[N], vec2[N], vec3[N] ;
    int i ;
    printf("Enter values into vector 1 : \n");
    for (i=0 ; i<N ; i++)
        scanf("%d",&vec1[i]);
    printf("Enter values into vector 2 : \n");
    for (i=0 ; i<N ; i++)
        scanf("%d",&vec2[i]);
    printf("vector 3 is : ");
    for (i=0 ; i<N ; i++)
    {
        vec3[i] = vec1[i] + vec2[i];
        printf("%d ",vec3[i]);
    }

    // תוספת שאלה 2
    int max_even_index ;
    // לולאה לאיתור מיקום הזוגי הראשון
    for (i=0 ; i<N && vec3[i] %2 ; i++) ;
    if (i==N) // אם לא נמצא ערך זוגי
        printf("-1") ;
    max_even_index = i ;
    i++ ;
    while (i<N) // לולאה לאיתור מיקום הזוגי הכי גדול
    {
        if (vec3[i]%2==0 && vec3[i]>vec3[max_even_index])
            max_even_index = i ;
        i++ ;
    }
    printf("%d",max_even_index) ;
}
```

// 3 תוספת שאלה

```
int cnt=0 ;
for (i=0 ; i<N ; i+=2)
    if (vec3[i]*vec3[i] == vec3[i+1] || vec3[i+1]*vec3[i+1] == vec3[i])
        cnt++ ;
printf("%d",cnt) ;
```

// 4 תוספת שאלה

```
int j, a ;
for (i=0, j=N-1 ; i<j ; i++,j--)
{
    a=vec3[i] ;
    vec3[i]=vec3[j] ;
    vec3[j]=a ;
}
}
```