

העברת מערכים לפונקציות

מסמך זה מתייחס להעברת מערכים מגודל ידוע מראש (קבוע) לפונקציות.

כדי להדגים העברת מערך דו מימדי לפונקציה, נתחיל בהגדרת קבועים שמגדירים את גודל מערך:

```
#define ROWS_NUM 2
```

```
#define COLS_NUM 2
```

כאשר רוצים להעביר לפונקציה כפרמטר מערך חד מימדי שנקרא arr בגודל ROWS_NUM, ניתן לעשות זאת בשתי דרכים שקולות:

```
int func(int arr[])
```

או

```
int func(int arr[ROWS_NUM])
```

שתי השיטות זהות מבחינת הקומפילר - הוא לא יתייחס לגודל המערך. ההבדל ביניהן הוא מבחינת קריאות הקוד בלבד – בשיטה השנייה המתכנת מצהיר על כוונתו שהפונקציה תעבוד עם מערכים בגודל מסוים בלבד.

בתוך הפונקציה צריך לשים לב לפנות לאינדקסים 0 עד ROWS_NUM - 1 בלבד כדי שלא תהיה חריגת זיכרון.

גם כאשר רוצים לכתוב פונקציה שמקבלת מערך דו מימדי כפרמטר, למשל פונקציה שמאפסת את כל תאי המערך, ניתן לבצע זאת בשתי דרכים שקולות:

```
void init_array(int arr[ROWS_NUM][COLS_NUM])
```

או

```
void init_array(int arr[][COLS_NUM])
```

שימו לב: במערכים דו מימדיים חובה לציין את מספר העמודות בכל שורה. אסור להעביר לפונקציה מערך עם מספר עמודות שונה ממה שהוגדר.

קריאה לפונקציה:

יש לקרוא לפונקציה ולספק את שם המערך, **ללא סוגריים מרובעים**. נדגים למשל קריאה לפונקציה שהוגדרה קודם.

```
int main()
```

```
{
```

```
    int arr[ROWS_NUM][COLS_NUM]; // contains garbage
```

```
    init_array(arr); // after this line, arr is all zeros
```

```
    return 0;
```

```
}
```

ב – C, כאשר מעבירים מערך כפרמטר לפונקציה (או כפי שנלמד בהמשך – מצביע), שינוי תוכן המערך בתוך הפונקציה משנה גם את תוכן המערך שהועבר כפרמטר (בניגוד לפרמטרים מטיפוסים אחרים ששינויים בתוך הפונקציה לא ישנו אותם מחוץ לפונקציה). לדוגמא:

```
void init_array(int arr[ROWS_NUM][COLS_NUM])
{
    for (int i = 0; i < ROWS_NUM; i++)
    {
        for (int j = 0; j < COLS_NUM; j++)
        {
            arr[i][j] = 0;
        }
    }
}

void init_int(int x)
{
    x = 0;
}

int main()
{
    int arr[ROWS_NUM][COLS_NUM]={{1,2},{3,4}};
    int x=5;
    init_array(arr); // after this line, arr is all zeros
    init_int(x); // after this line, x still 5
    printf("arr[0][0]=%d\n", arr[0][0]); //prints 0
    printf("x=%d\n", x); //prints 5
    return 0;
}
```