

## איך להקצות דינאמית מערך דו-מימדי בשפת C?

יש שתי דרכים להגדיר מערך דו-מימדי בגודל  $m \times n$  בצורה דינאמית.

דרך אחת, היא פשוט להגדיר מערך חד-מימדי בגודל  $m \cdot n$  ולהתייחס אליו כמערך דו-מימדי.

אז מטריצה כזו: 
$$\begin{bmatrix} 2 & 5 \\ -1 & 8 \\ 0 & 3 \end{bmatrix}$$
 תיוצג על ידי מערך חד-מימדי כזה: 
$$\begin{bmatrix} 2 & 5 & -1 & 8 & 0 & 3 \end{bmatrix}$$

נגדיר את המערך המתאים בצורה הבאה:

```
int* array = (int*)malloc(m * n * sizeof(int));
```

ואז כדי לגשת לתא ה- $j$ , במטריצה, ניגש לתא ה- $n*i+j$  במערך. יש  $m$  שורות ו- $n$  עמודות במטריצה, כלומר  $n$  איברים בכל שורה. לכן, כדי להגיע לתא ה- $j$ , במטריצה, נצטרך "לדלג" על  $i$  שורות (שהם  $i*n$  איברים), ועל עוד  $j$  איברים. ככה נגיע בדיוק לאיבר המבוקש.

```
array[n*i+j]  
m[i][j] שקול ל
```

לבסוף לא נשכח לשחרר את המערך ע"י `free(a)`.

דרך נוספת להגדיר מערך דו-מימדי היא בצורה הבאה:

מערך דו-מימדי הוא מערך של מערכים. אם נסתכל על מטריצה, אז כל שורה בה היא מערך של מספרים, וכל המטריצה בשלמותה היא מערך של שורות (מערך של מערכים).  
בשביל להגדיר דינאמית מערך דו-מימדי, נצטרך תחילה להקצות דינאמית מערך של שורות. כל שורה היא מערך, אז בעצם כשנקצה מערך של שורות, אנחנו נקצה מערך של מצביעים.  
אחר כך עבור כל מצביע כזה, נרצה שהוא יצביע למערך המייצג שורה במטריצה. לשם כך אנחנו נבצע הקצאה דינמית עבור השורה הזו (שהיא מערך בפני עצמה).

דוגמה לקוד שמקצה דינאמית מטריצה בגודל  $m \times n$ :

```
int** array = (int**)malloc(m * sizeof(int*));
if (array == NULL)
    return -1;
for (i=0; i<m; i++)
{
    array[i] = (int*)malloc(n * sizeof(int));
    if (array[i] == NULL)
        return -1;
}
```

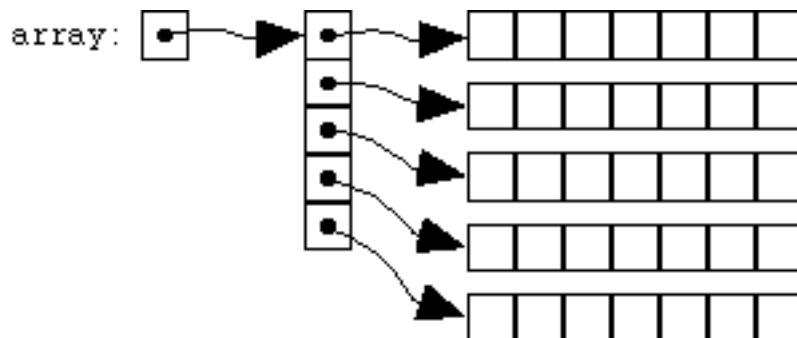
\* שימו לב שיש m שורות ו-n עמודות במטריצה, כלומר n איברים בכל שורה.

לאחר ביצוע קטע הקוד הנ"ל המטריצה הוקצתה דינאמית, ואפשר להשתמש בה כרגיל, למשל:

```
for (i=0; i<m; i++)
    for (j=0; j<n; j++)
        array[i][j] = 0;
```

לבסוף כדי לשחרר את הזיכרון, נצטרך לבצע free קודם לשורות ואחר כך למטריצה כולה.

```
for (i=0; i<m; i++)
{
    free(array[i]);
}
free(array);
```



שימו לב שהטיפוס של המטריצה הוא `int**`. טיפוס זה הוא מצביע למשתנה מסוג `int*`, כלומר מצביע למצביע.

למה אנחנו צריכים מצביע למצביע?

כל שורה במטריצה היא מערך, כלומר `int[]`. אבל בגלל שכל שורה מוקצת דינאמית, הטיפוס שלה הוא `int*`.

לכן מערך של איברים מטיפוס `int*` הוא יהיה מטיפוס `int**`. כך הוא מצביע לתחילת מערך של איברים מטיפוס `int*`.