

CURS 03 – PP

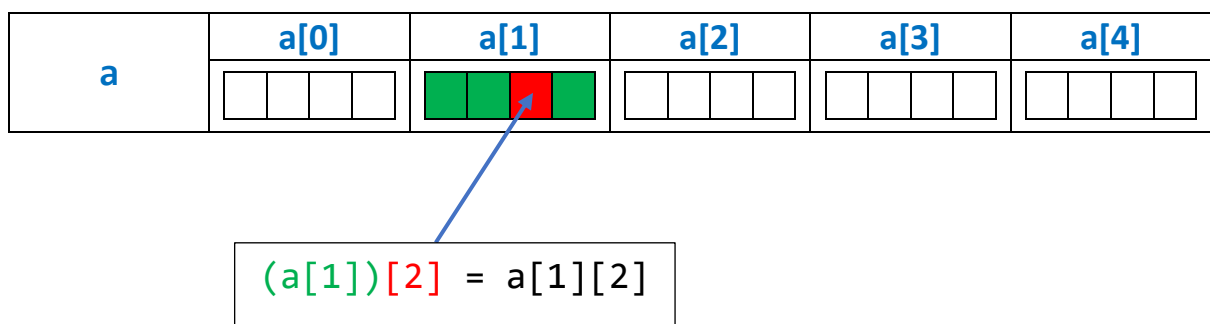
ALOCAREA DINAMICĂ A TABLOURILOR BIDIMENSIONALE

Tablou bidimensional = un tablou unidimensional ale cărui elemente sunt tablouri unidimensionale (liniile sale)

Exemplu:

```
int a[5][4];
```

$a[i]$ = un tablou unidimensional reprezentând linia i a tabloului bidimensional



Observație: Elementele unui tablou bidimensional ocupă o zonă continuă de memorie!!!

Exemplu:

```
int a[5][4];
```

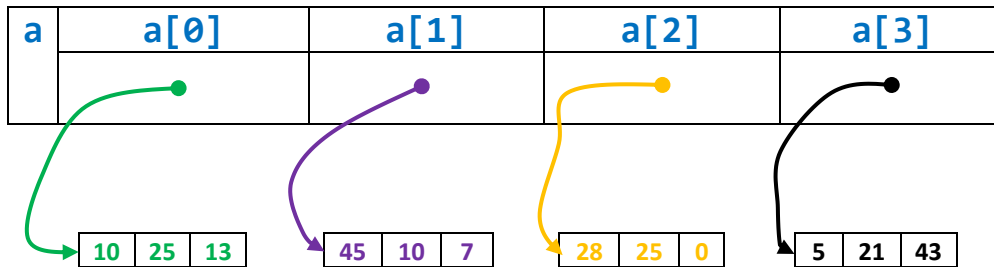
$m = 4$ linii și $n = 3$ coloane

$$a = \begin{pmatrix} 10 & 25 & 13 \\ 45 & 10 & 7 \\ 28 & 25 & 0 \\ 5 & 21 & 43 \end{pmatrix}$$

	a[0]				a[1]				a[2]				a[3]				a[4]			
a	10	25	13	?	45	10	7	?	28	25	0	?	5	21	43	?	?	?	?	?

? = valoare reziduală

ALOCAREA DINAMICĂ A UNUI "TABLOU BIDIMENSIONAL"



m = numărul de linii (4)

n = numărul de coloane (3)

a[i] = adresa de început a unei zone de memorie din HEAP unde a fost alocată linia i tabloului bidimensional

a = adresa de început a unui tablou unidimensional de adrese = adresa unei adrese = pointer dublu

Alocarea dinamică a unui "tablou bidimensional"

```
//tabloul bidimensional este, de fapt, un pointer dublu  
int **a;
```

```
//alocăm tabloul unidimensional "a" cu m elemente  
//de tip int*  
a = (int**)malloc(m * sizeof(int*));
```

```
//alocăm fiecare linie, adică un tablou unidimensional cu n  
elemente de tip int  
for(i = 0; i < m; i++)  
    a[i] = (int*)malloc(n * sizeof(int));
```

Eliberarea zonei de memorie alocată dinamic pentru un "tablou bidimensional"

```
//eliberăm zona de memorie alocată fiecărei linii  
for(i = 0; i < m; i++)  
    free(a[i]);
```

```
//eliberăm zona de memorie alocată tabloului bidimensional  
free(a);
```