Fondamenti di Programmazione (A)

12 - Strutture dati (Array bi-dimensionali)

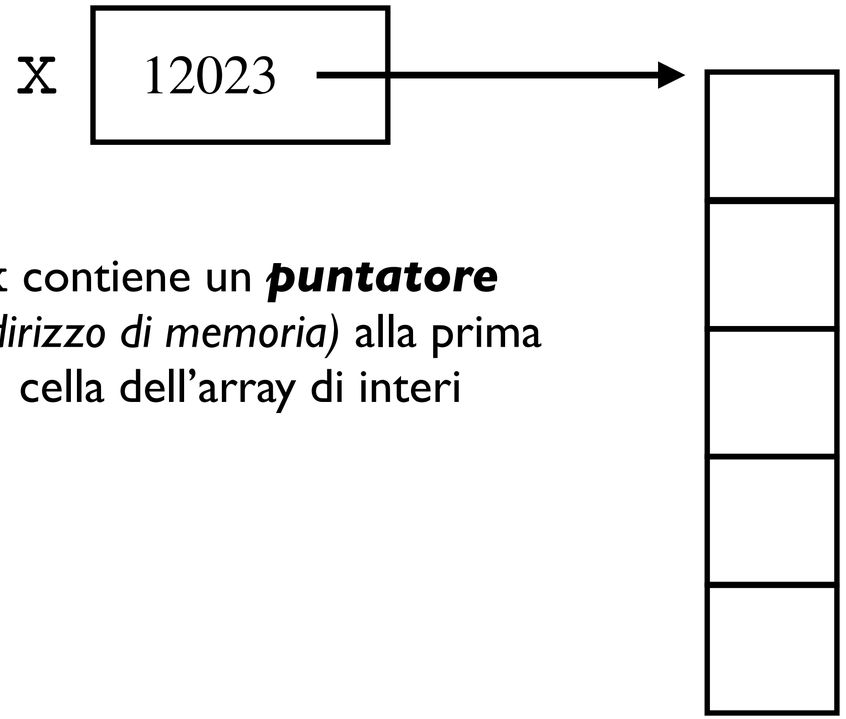
Vincenzo Arceri - Università degli Studi di Parma - vincenzo.arceri@unipr.it

Puntate precedenti

- Array (monodimensionali)
  - Array statici
  - Array semi-dinamici

Array mono-dimensionali

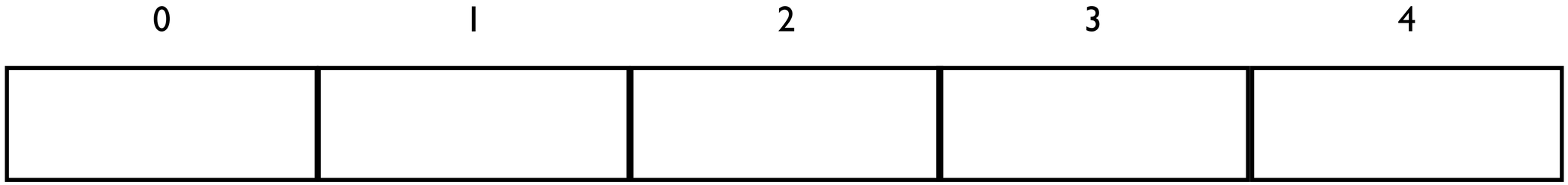
```
int x[5];
```



`x` contiene un ***puntatore*** *(indirizzo di memoria)* alla prima cella dell'array di interi

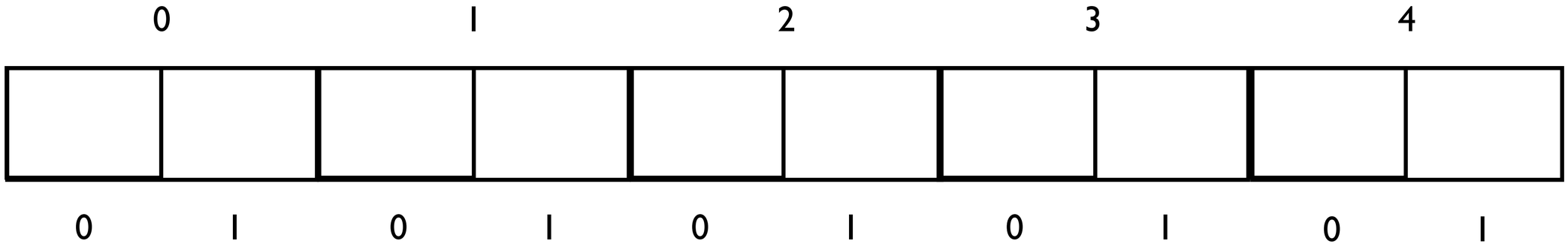
Array bi-dimensionali

- Un array bi-dimensionale $m \times n$ è un **array monodimensionale** di m elementi tale che ogni elemento è, a sua volta, un **array monodimensionale** di n elementi
- Esempio: array bi-dimensionale 5 x 2



Array bi-dimensionali

- Un array bi-dimensionale $m \times n$ è un **array monodimensionale** di m elementi tale che ogni elemento è, a sua volta, un **array monodimensionale** di n elementi
- Esempio: array bi-dimensionale 5 x 2



Array bi-dimensionali

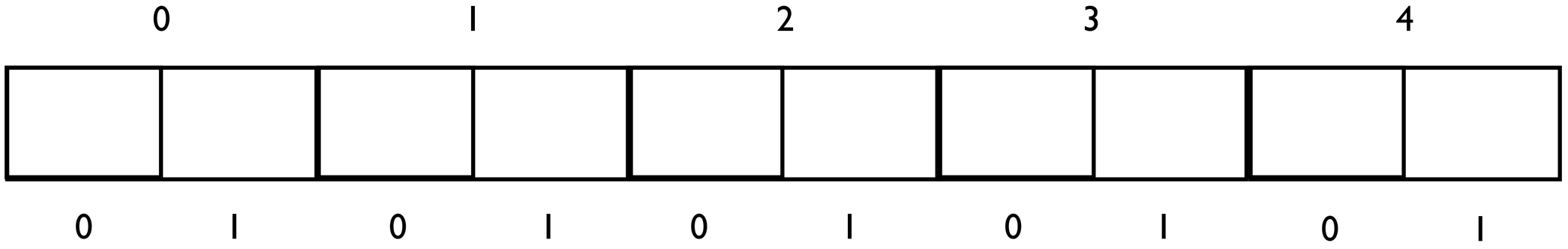
In C++

$$t\ \mathtt{A}[m][n];$$

- t è un tipo (a sua volta può essere strutturato) e indica il tipo che ogni cella dell'array può contenere
- m e n sono espressioni intere
- $\mathtt{A}$ è un identificatore

Array bi-dimensionali

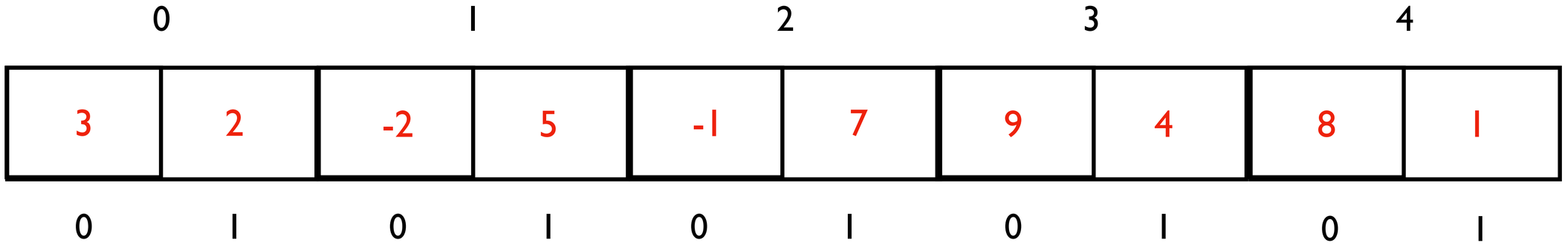
Esempio in C++




```
int  x[5][2];
```

Array bi-dimensionali

Esempio in C++

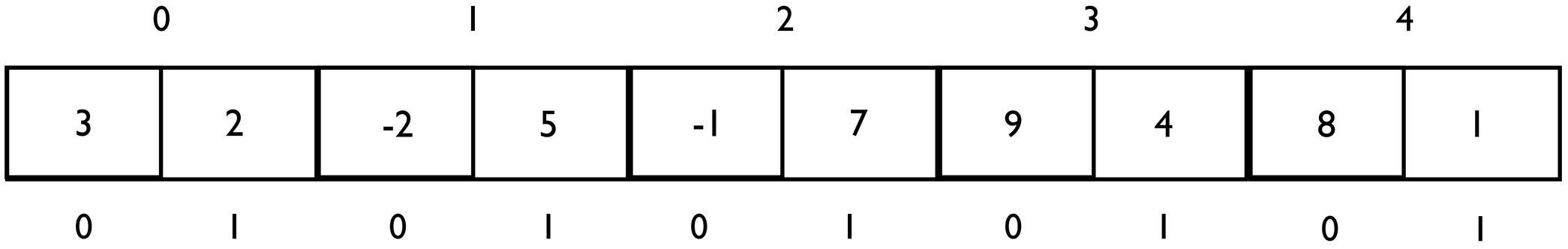



```
int  x[5][2];
```

E se voglio inizializzato?

Array bi-dimensionali

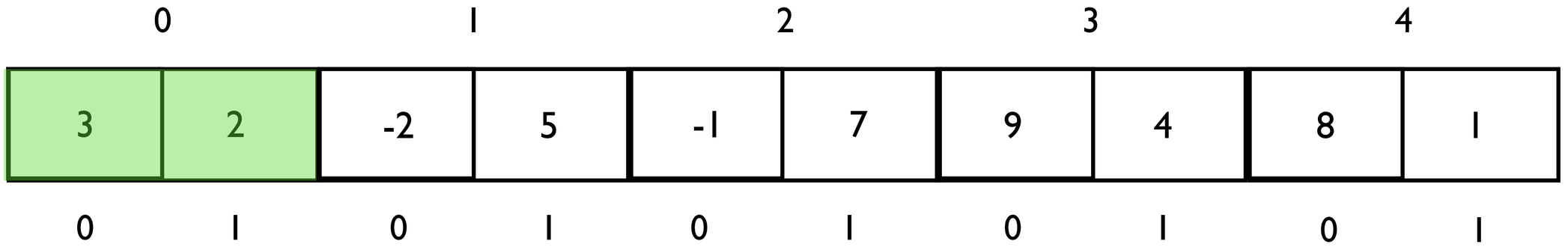
Esempio in C++




```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

Array bi-dimensionali

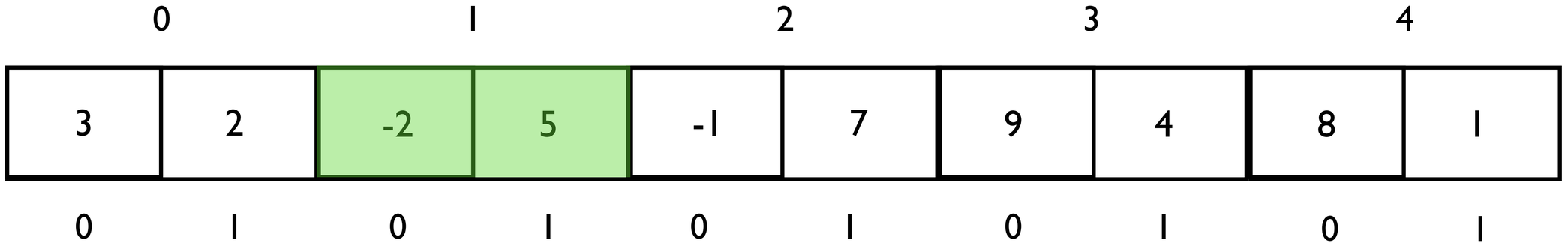
Esempio in C++




```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

Array bi-dimensionali

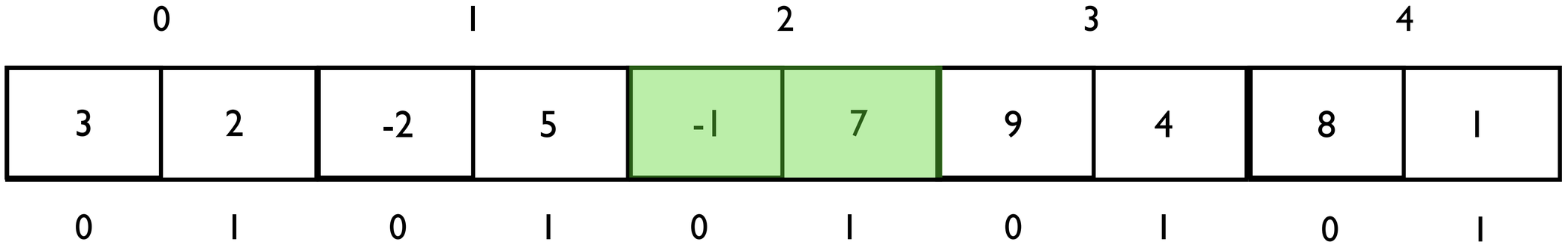
Esempio in C++




```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

Array bi-dimensionali

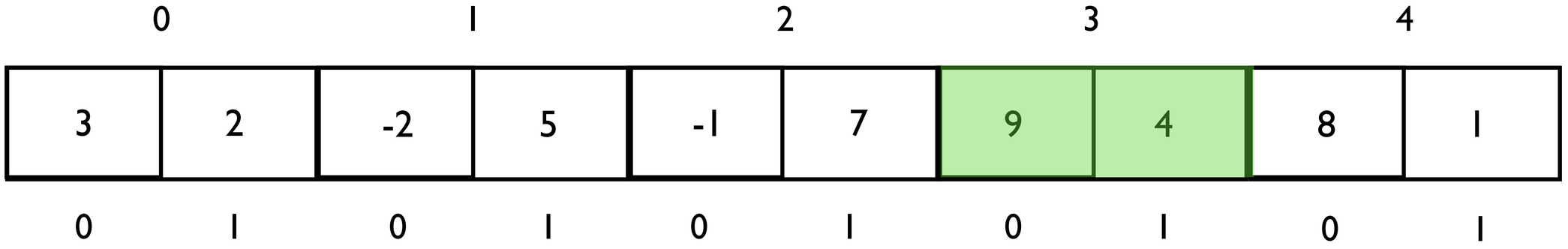
Esempio in C++




```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

Array bi-dimensionali

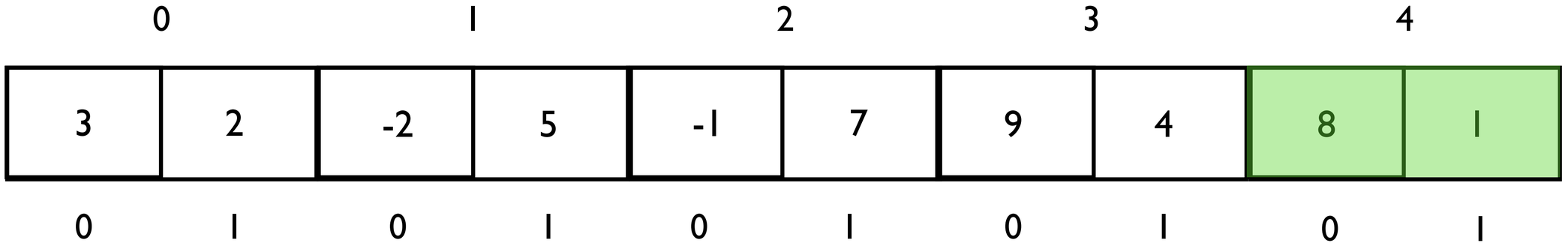
Esempio in C++




```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

Array bi-dimensionali

Esempio in C++




```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

Array bi-dimensionali

Selezione

- Dato un array bi-dimensionale, è possibile *accedere* (in lettura e scrittura) ai suoi elementi tramite **accesso diretto**

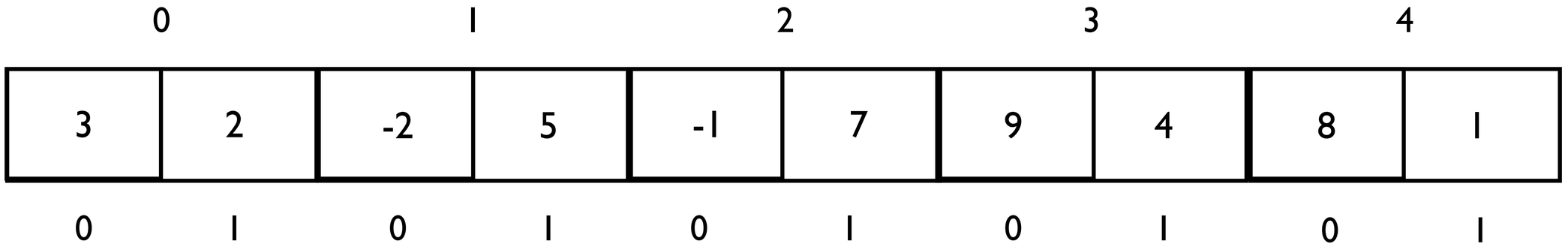
$\texttt{A}[exp_1][exp_2]$

`A` è un array bi-dimensionale
exp_1 e exp_2 sono espressioni compatibili col tipo `int`

Array bi-dimensionali

Esempio di selezione (lettura)

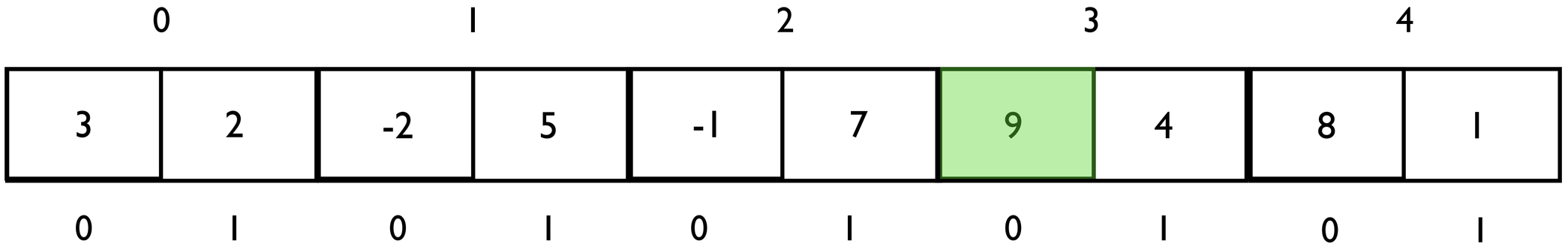
```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```



Array bi-dimensionali

Esempio di selezione (lettura)

```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

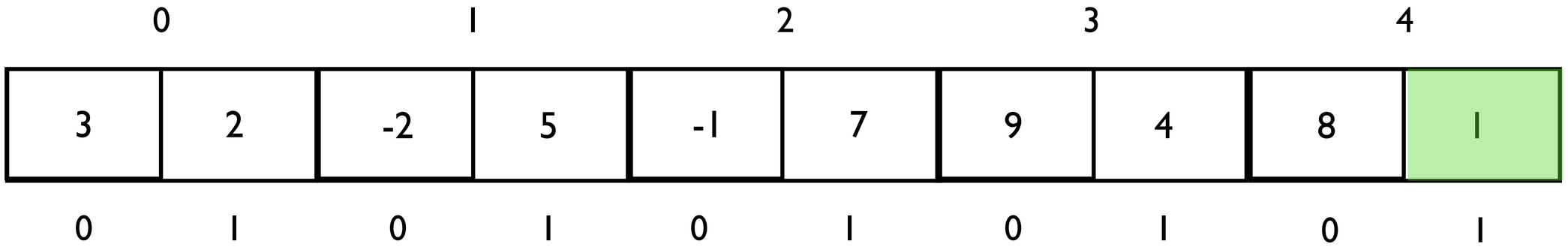


x[3][0]

Array bi-dimensionali

Esempio di selezione (lettura)

```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```

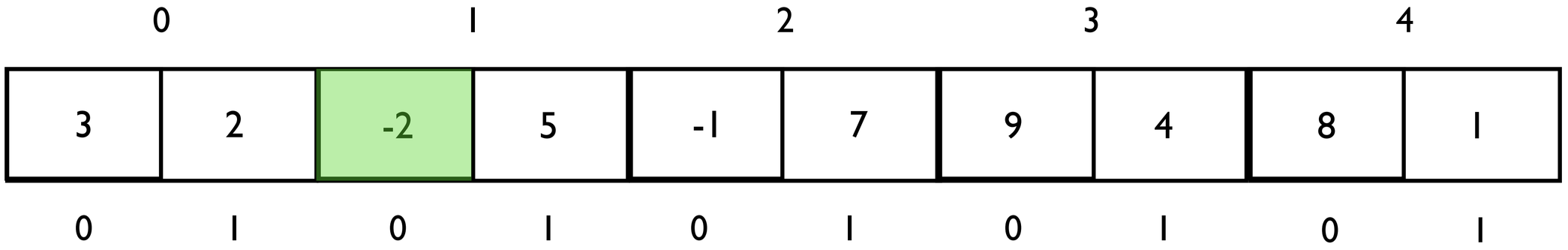


x[4][1]

Array bi-dimensionali

Esempio di selezione (lettura)

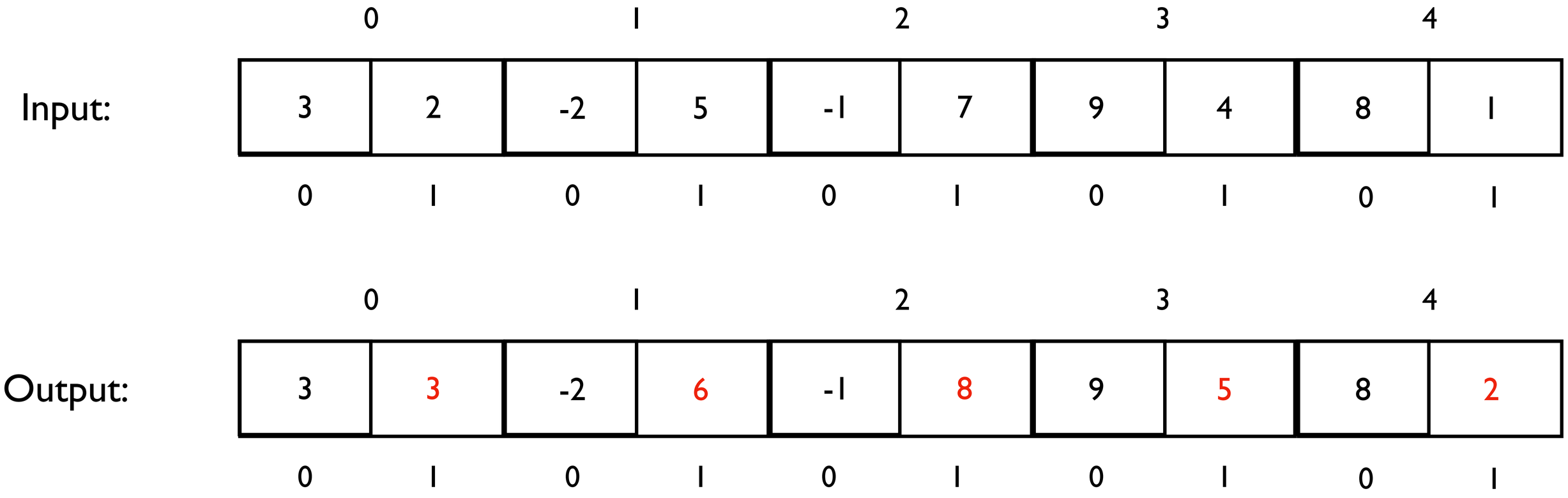
```
int x[5][2] = { {3,2}, {-2,5}, {-1,7}, {9,4}, {8,1} };
```



x[1][0]

Array bi-dimensionali

Esempio di selezione (scrittura)



- Problema: dato in input l'array-bidimensionale in figura, incrementare la seconda cella di ciascun array in posizione $i \in [0,4]$

Array bi-dimensionali

Allocazione in memoria

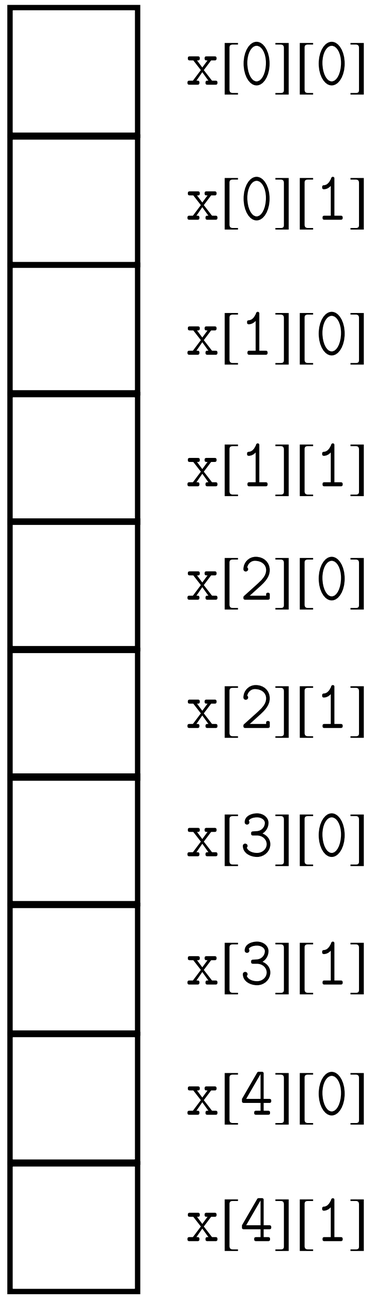
```
int x[5][2];
```

- Un array bi-dimensionale $m \times n$ di tipo t alloca m array **consecutivi**, ciascuno di n elementi di tipo t

Array bi-dimensionali

Allocazione in memoria

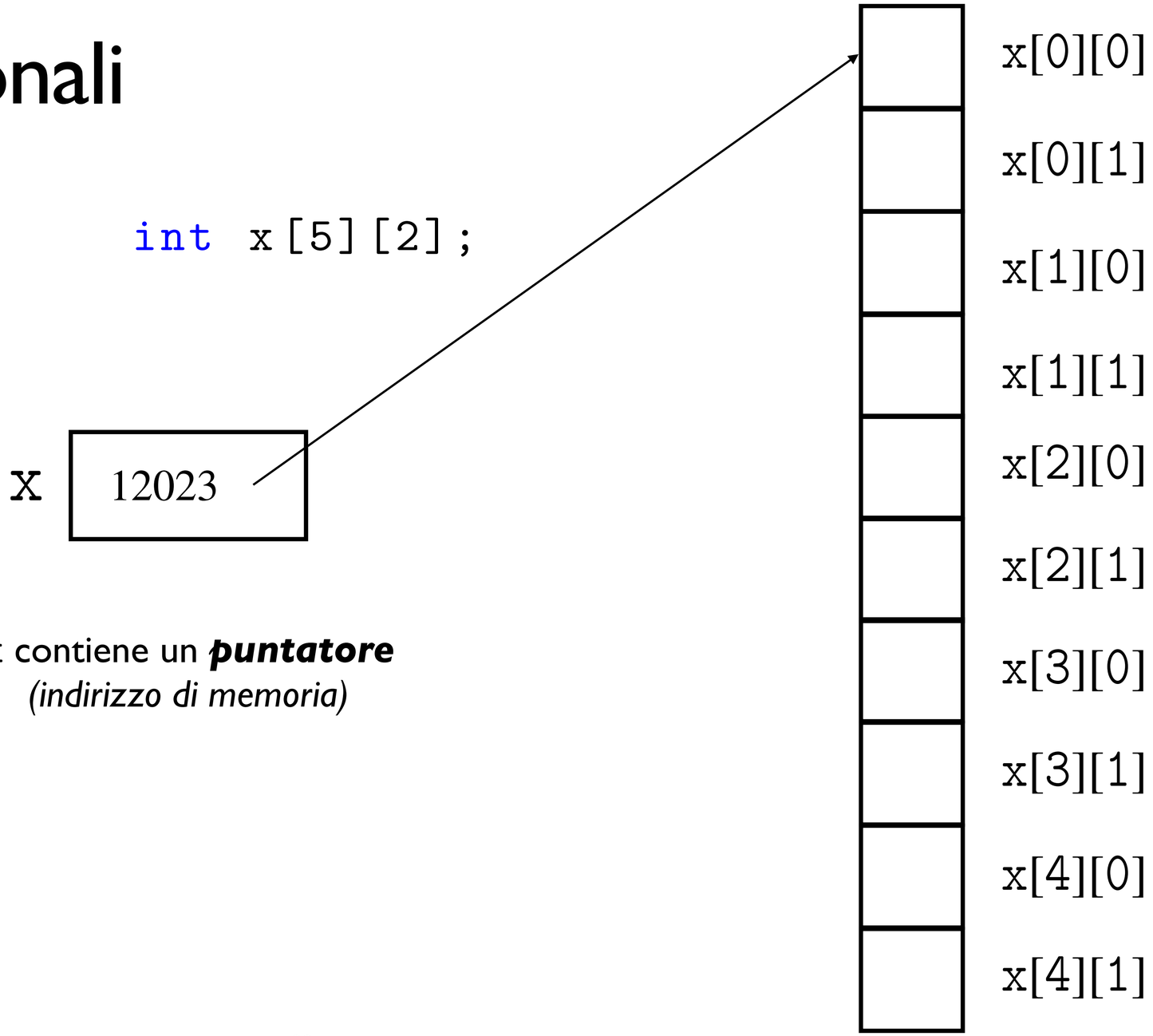
```
int  x[5][2];
```



Array bi-dimensionali

Allocazione in memoria

```
int  x[5][2];
```



`x` contiene un ***puntatore***
(indirizzo di memoria)

Array bi-dimensionali

Matrici

- Gli array bi-dimensionali si prestano bene all'implementazione delle *matrici*: dati m e n interi positivi, una matrice $m \times n$ ha la seguente forma

$$a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

Array bi-dimensionali

Matrici

$$a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

- Data una matrice $m \times n$, m è il numero di righe, n è il numero di colonne, a_{ij} è l'elemento ij della matrice a

Array bi-dimensionali

Matrici

$$a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

- Data una matrice $m \times n$, m è il numero di righe, n è il numero di colonne, a_{ij} è l'elemento ij della matrice a
- i-esima **riga** di a: $a_i = (a_{i1}, a_{i2}, a_{i3}, \ldots, a_{in})$

Array bi-dimensionali

Matrici

$$a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

- Data una matrice $m \times n$, m è il numero di righe, n è il numero di colonne, a_{ij} è l'elemento ij della matrice a
- i-esima **riga** di a: $a_i = (a_{i1}, a_{i2}, a_{i3}, \dots, a_{in})$
- i-esima **colonna** di a: $a_i = (a_{1i}, a_{2i}, a_{3i}, \dots, a_{mi})$

Array bi-dimensionali

Matrici

Matrice 3 x 2

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 1 & 2 \\ 5 & 7 \\ 3 & -6 \end{bmatrix} \end{array}$$

Dichiarazione in C++

```
int x[3][2] = { {1,2}, {5,7}, {3,-6} };
```

Rappresentazione grafica in C++

	0	1
0	1	2
1	5	7
2	3	-6

Matrici

Esercizi

- Leggere una matrice $m \times n$, con m, n e i suoi elementi chiesti all'utente
- Stampare la matrice letta

Matrici

Esercizi

- Leggere due matrici $m \times n$, con m, n e i suoi elementi chiesti all'utente, e calcolare la somma delle due matrici

$$\begin{bmatrix} -1 & 3 & -2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 2 & -5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 3 & 3 \\ 3 & -5 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

Matrici

Esercizi

- Leggere una matrice $m \times n$, con m, n e i suoi elementi chiesti all'utente, e stampare la sua matrice trasposta

$$A = \begin{pmatrix} 2 & 4 & 8 \\ 3 & 2 & 0 \\ 5 & 3 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad A^T = \begin{pmatrix} 2 & 3 & 5 & 0 \\ 4 & 2 & 3 & 1 \\ 8 & 0 & 1 & 0 \end{pmatrix}$$

Array k-dimensionali

- Dichiarazione di un array k-dimensionale

$$\texttt{t a[n}_1\texttt{][n}_2\texttt{]...[n}_k\texttt{];}$$

- Accesso ad un array k-dimensionale

$$\texttt{a[n}_1\texttt{][n}_2\texttt{]...[n}_k\texttt{]}$$