



UNIVERSITÀ
degli STUDI
di CATANIA

Array

Corso di programmazione I

Corso di Laurea Triennale in Informatica

Prof. Giovanni Maria Farinella

Web: <http://www.dmi.unict.it/farinella>

Email: gfarinella@dmi.unict.it

Dipartimento di Matematica e Informatica

1. Introduzione agli array
2. Array: lettura, scrittura e loop
3. Array a più dimensioni
4. Esercizi

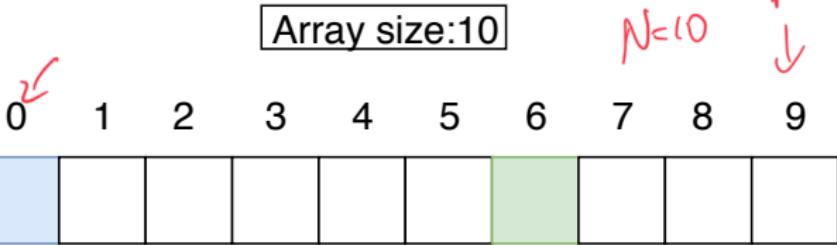
2

"2"

Introduzione agli array

Definizione di array monodimensionale

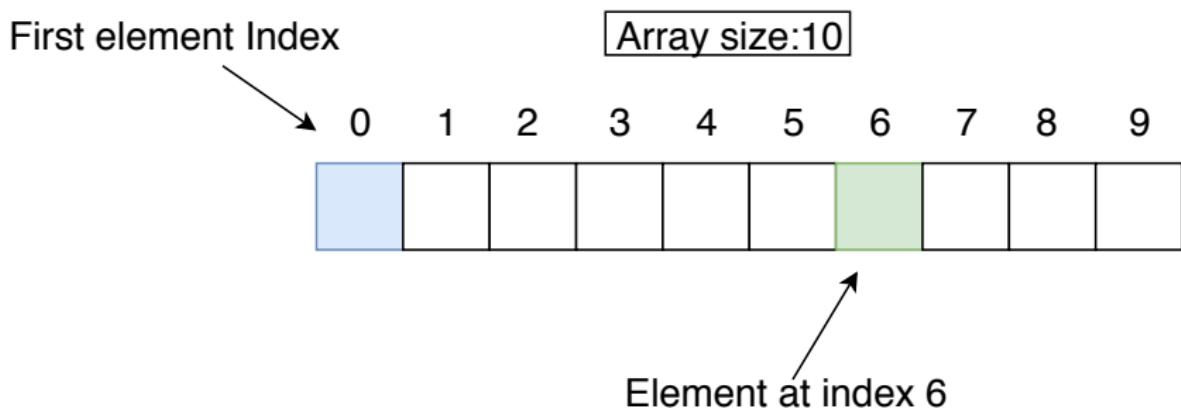
First element Index



ARRAY CON
N CELLE HA
INDICI CHE
VANNO DA 0
A $N-1$

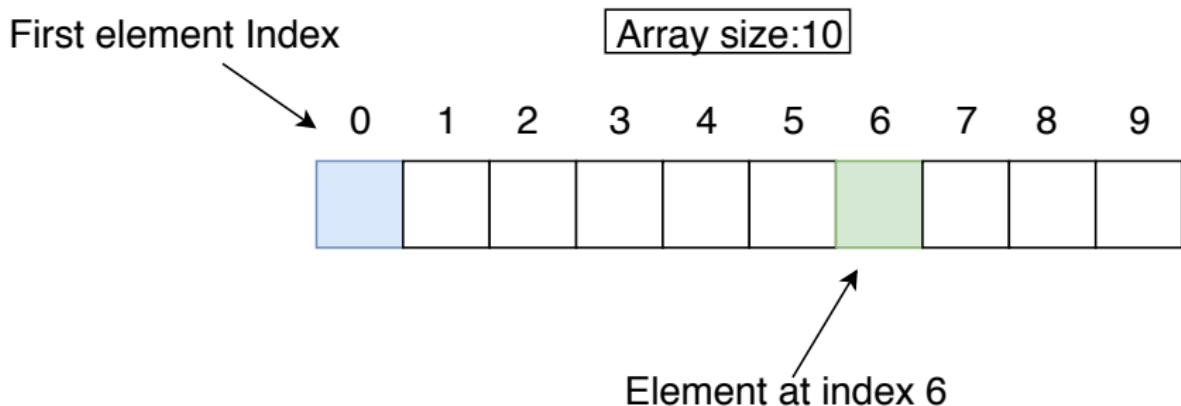
Un array e' una struttura dati omogenea, che funge da contenitore di elementi dello stesso tipo.

Definizione di array monodimensionale



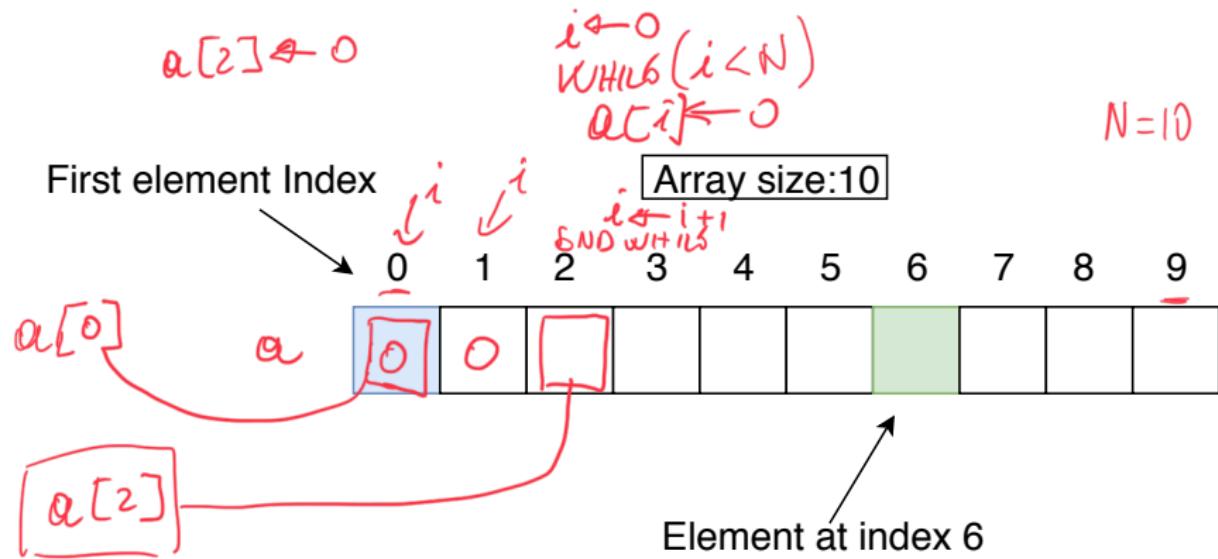
Gli elementi (o celle) sono identificati da un indice numerico.

Definizione di array monodimensionale



Gli indici partono da 0 (è così nel linguaggio C/C++) oppure da 1.

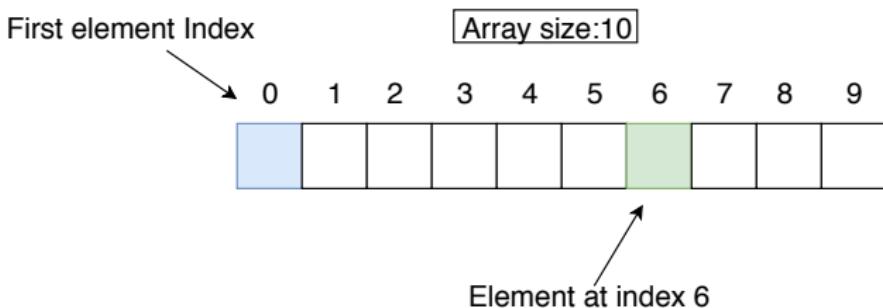
Definizione di array monodimensionale



Il valore delle celle sarà accessibile in lettura e scrittura mediante i)
nome dello array, ii) parentesi quadre e iii) indice. ES: $a[3]$.

Array: lettura, scrittura e loop

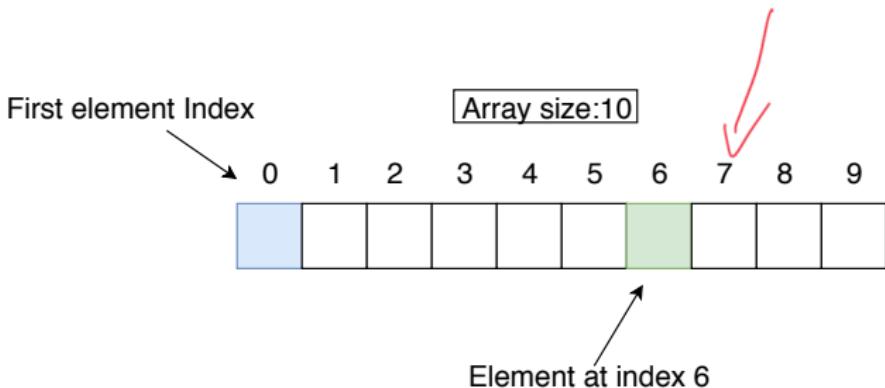
Uso degli array: lettura e scrittura



Assegna all'elemento di indice 5 il valore 15

a[5] ← 15;

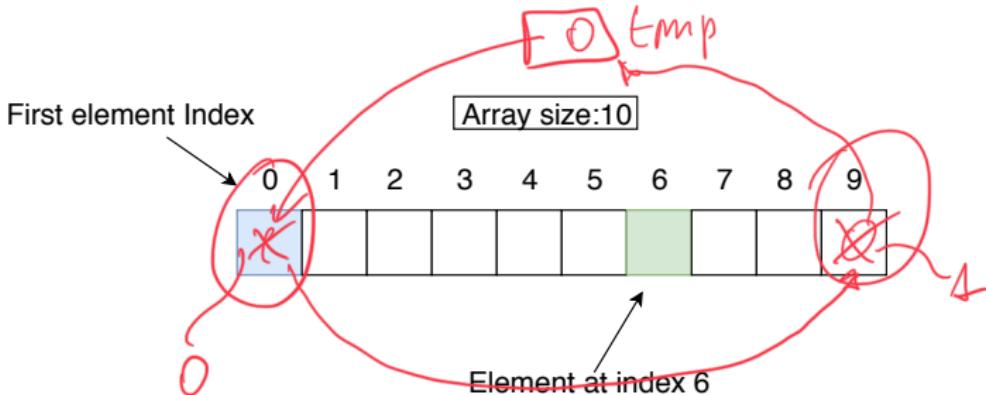
Uso degli array: lettura e scrittura



Scrivi in Y il valore dell'elemento dello array a di indice 7

$\underline{Y \leftarrow a[7];}$

Uso degli array: lettura e scrittura



Scambia il primo elemento con l'ultimo

```
tmp ← a[9];  
a[9] ← a[0];  
a[0] ← tmp;
```

—]

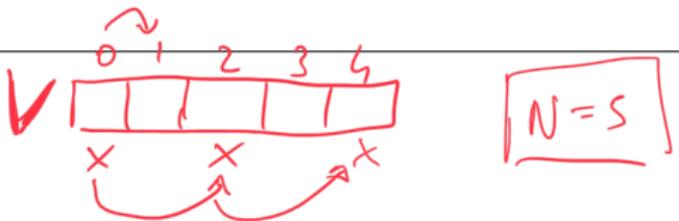
$a[9] \leftrightarrow a[0]$
 $a[0] \leftrightarrow a[9]$

Cicli su array monodimensionali

Sia V un array di N elementi.

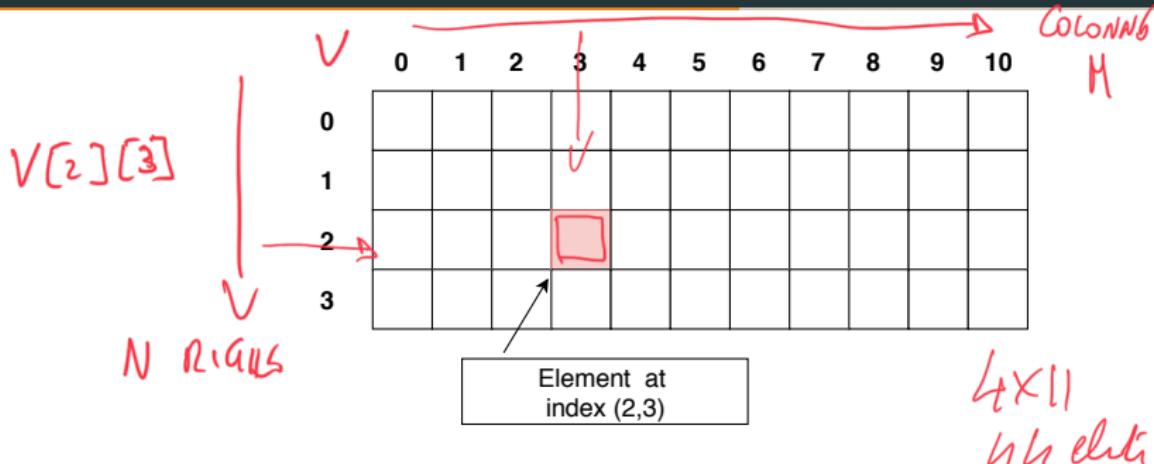
Descrivere un algoritmo che calcoli la **somma degli elementi di indice pari** ($0, 2, 4, \dots$).

```
1 Inizio
2 Leggi N ←
3 Leggi V ←
4 S ← 0 ←
5 i ← 0
6 While( i < N ) Do
7   S ← S + V[i] ←
8   i ← i + 2 ←
9 End While
10 Stampa S
11 Fine
```



Array a più dimensioni

Array bidimensionali

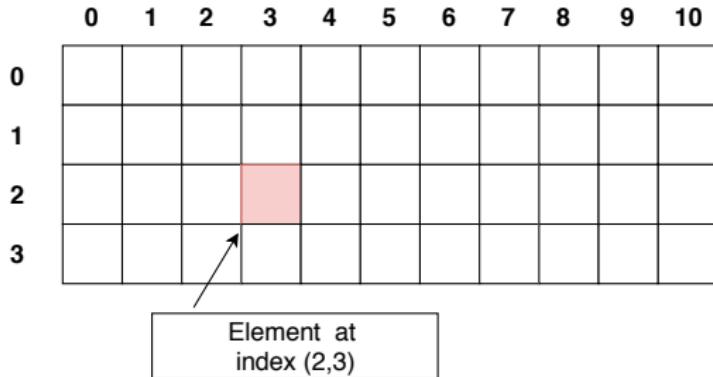


Un **array bidimensionale** o matrice è una struttura dati omogenea che si estende su due dimensioni (graficamente una tabella).

Un array bidimensionale di dimensioni $N \times M$ è composto di $N \cdot M$ elementi

ES: Un array o matrice 4×5 è composto di $4 \cdot 5 = 20$ elementi

Array bidimensionali

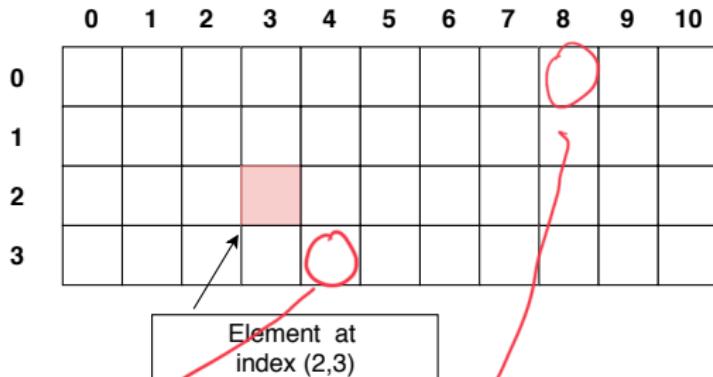


L'accesso ai suoi elementi avviene tramite **una coppia di indici**.

Pone il valore di Y nello elemento di a di indici $(1, 2)$

$a[1][2] \leftarrow Y;$

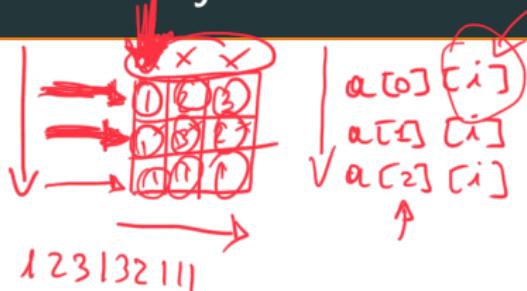
Array bidimensionali



Moltiplica i valori in corrispondenza dei indici (3,4) e (0,8) e pone il risultato nella variabile X

$X \leftarrow a[3][4] \cdot a[0][8];$

Cicli su array bidimensionali



Sia V un array bidimensionale di $N \times M$ elementi (N righe, M colonne).

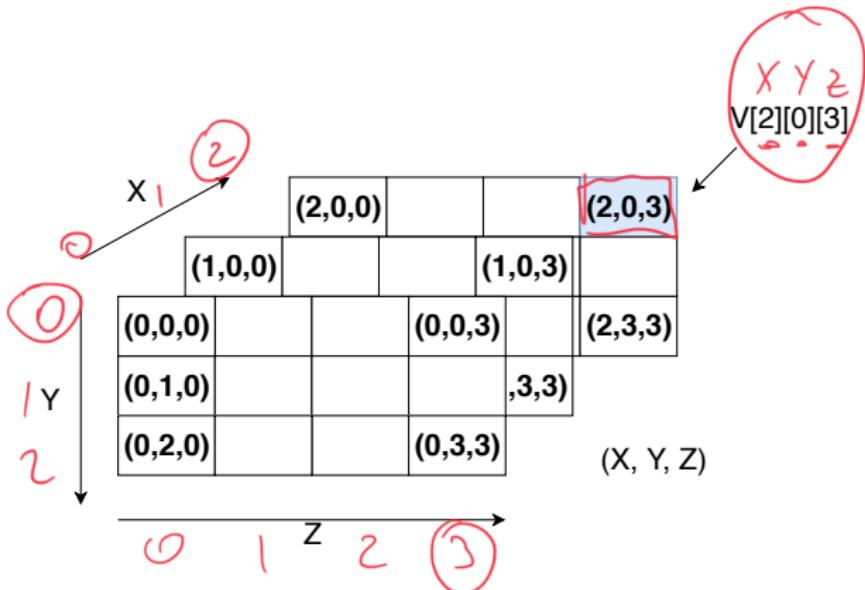
Descrivere un algoritmo che calcoli la somma di tutti gli elementi dello array o matrice.

$\text{WHILE } (j < N)$
 $\quad \text{WHILE } (i < M)$
 $\quad \quad a[i][j] \text{ USA}$
 $\quad \quad i \leftarrow i + 1$
 $\quad \text{END WHILE}$
 $\quad j \leftarrow j + 1$
 $\quad i \leftarrow 0$
 END WHILE

i colonna
 j righe

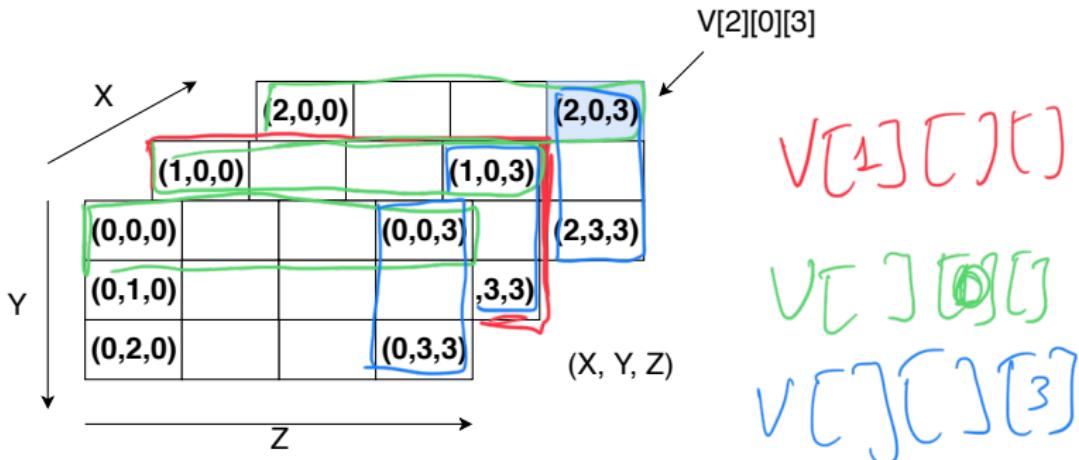
1 **Inizio**
2 Leggi N ;
3 Leggi M ;
4 Leggi V ;
5 $S \leftarrow 0$; *i righe*
6 $i \leftarrow 0$;
7 **While** ($i < N$) **Do** *j colonne*
8 $j \leftarrow 0$;
9 **While** ($j < M$) **Do**
10 $S \leftarrow S + V[i][j]$;
11 $j \leftarrow j + 1$
12 **End While** *i+1*
13 $i \leftarrow i + 1$
14 **end while**
15 Stampa S ;
16 **Fine**

Array k-dimensionali. $k = 3$



Un array k-dimensionale o matrice è una struttura dati omogenea che si estende su k dimensioni

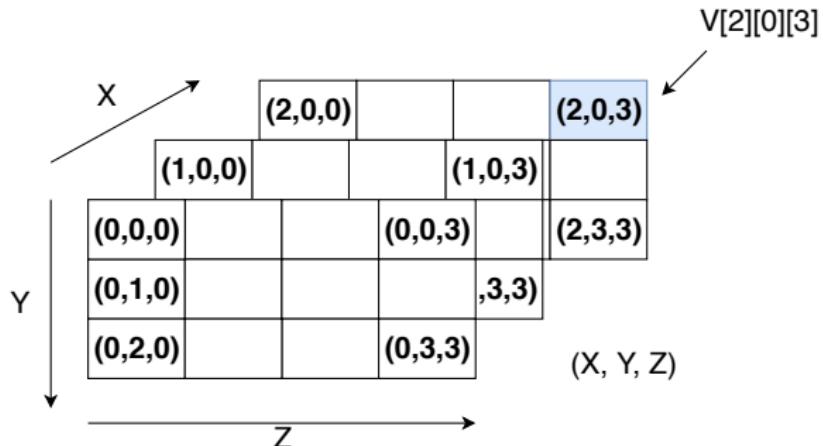
Array k-dimensionali. $k = 3$



La rappresentazione grafica di un array tridimensionale permette di comprendere alcuni concetti di base.

In un array k-dimensionale, **fissando un valore per uno qualsiasi degli indici** si otterrà uno array $(k-1)$ -dimensionale.

Array k-dimensionali. $k = 3$



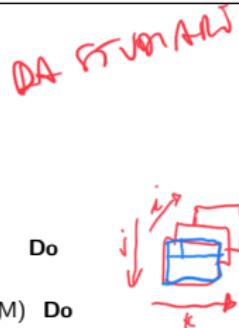
Quindi, con riferimento alla figura porre $x = 0$ equivale a manipolare un array bidimensionale di dimensioni 3×4 .

Cicli su array k-dimensionali (k=3)

Sia V un array tridimensionale di $N \times M \times O$ elementi.

Descrivere un algoritmo che calcoli **il prodotto di tutti gli elementi dello array**.

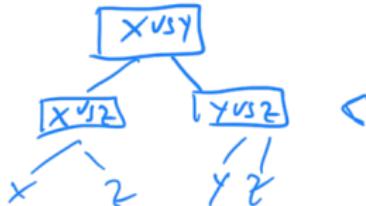
```
1 Inizio
2 Leggi N;
3 Leggi M;
4 Leggi O;
5 Leggi V;
6 S ← 1;
7 i ← 0;
8 While(i < N) Do
9   j ← 0;
10  While(j < M) Do
11    k ← 0;
12    While(k < O) Do
13      S ← S · V[i][j][k];
14      k ← k + 1;
15    End While
16    j ← j + 1;
17  End While
18  i ← i + 1;
19 End While
20 Stampa S;
21 Fine
```



Esercizi

Esercizi

Homework



A1

Descrivere un algoritmo in notazione NLS che prende in input tre numeri X, Y, Z.

L'algoritmo deve calcolare e stampare il **minore dei tre numeri.**

A2

Descrivere un algoritmo in notazione NLS che prende in input un array A di N elementi.

L'algoritmo calcola e stampa il **massimo ed il minimo valore tra tutti gli elementi di A.**



$$N=3 \quad M=4 \quad 0 \ 2 \ 4 \ 5 \ 7.9 //$$

A3

Descrivere un algoritmo in notazione NLS che prende in input un numero $N > 0$ ed un numero $M > 0$. L'algoritmo opera nel seguente modo:

- stampa i **primi N numeri pari**, a partire da 0.
- stampa i **successivi M numeri dispari**.

Ad esempio, per $N=5$, $M=4$ lo output sarà:

0 2 4 6 8 9 11 13 15

Lega N
Lega M

A4

Descrivere un algoritmo in notazione NLS che prende in input un'array A di dimensione N ed un numero p. Si assume $0 < p < N$.

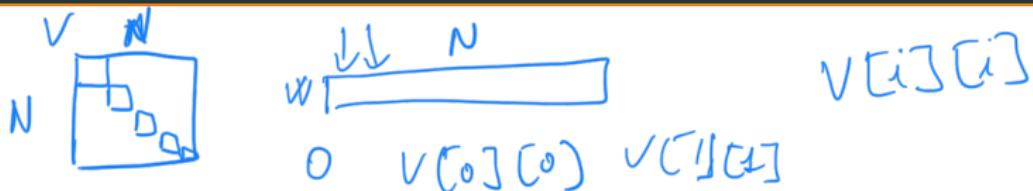
L'algoritmo esegue le seguenti operazioni:

- calcola il **massimo valore** degli elementi dello array A che hanno **indice minore o uguale a p**.
- calcola il **minimo valore** degli elementi dello array A con **indice maggiore di p**.
- calcola e stampa la **media aritmetica** tra i due valori calcolati in precedenza.

A5

Descrivere un algoritmo in notazione NLS che prende in input un array A di N elementi e due numeri a e b. Si assuma $a < b$. L'algoritmo dovrà stampare il **numero di elementi dello array** (sia $A[i]$ il generico elemento di A) **tali che $a \leq A[i] \leq b$** .

Esercizi



A6

Descrivere un algoritmo in notazione NLS che prende in input una matrice o array bidimensionale V di dimensione $N \times N$.

L'algoritmo costruisce un nuovo array W di dimensione N che contiene **gli elementi della diagonale principale di V** .