



UNIVERSITY OF TRENTO - Italy

Laboratorio 5

Pierluigi Roberti
Carmelo Ferrante

DISI – aa 2024/2025
Università degli Studi di Trento
pierluigi.roberti@unitn.it

Da Ricordare

- Gli array sono sequenze di celle consecutive omogenee
- Dichiarazione:

➤ `int vect[N];` // dove *N* è un numero intero precedentemente valorizzato

➤ `int vect[5] = {6,2,5,4,2};`

➤ `int vect[] = {6,2,5,4,2};`

6	2	5	4	2
---	---	---	---	---

- Gli elementi sono numerati da 0 fino a N-1 (non da 1 a N)
- Usare gli elementi degli array:
 - `vect[1] = 12;` // Assegnare ad un elemento dell'array

`int s = vect[4] * 7;` // Leggere il valore di un elemento

6	<u>12</u>	5	4	2
---	-----------	---	---	---

Direttiva #define

- Dichiarare
 - Label | Etichetta
 - Valore costante per tutto il programma

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
int main( ){
    . . .
}
```

- Fuori da main

Numeri interi casuali

- Generare numeri casuali:
- **rand()**
genera un numero intero compreso tra 0 e un valore massimo
RAND_MAX (2.147.483.647)
- Per generare un valore compreso tra 0 e $M - 1$:
rand() % M;
- Per generare un valore compreso tra **min** e **max**
rand() % (max - min + 1) + min;
- **srand(time(0))**
 - re-inizializza il generatore di numeri pseudo-casuali
 - da usare una sola volta nel programma
 - **#include <time.h>**

Esempio

- Generare numero casuale:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```
int main( ){
    srand(time(0));

    int i = rand( ) ;
    printf("%d", i);
    system("PAUSE");
    return 0;
}
```

Da Sapere - Vettori

```
int v[MAX];
```

effettiva dimensione
del vettore

- Inizializzare un vettore

```
for(i=0 ; i<len ; i++) {  
    v[i] = rand() % (280 - 4 + 1) + 4;  
}
```

- oppure

```
for(i=0 ; i<len ; i++){  
    printf("\nInserisci numero: ");  
    scanf("%d", &v[i]);  
}
```

len<=MAX

Da Sapere - Vettori

- Stampa contenuto di un vettore

```
for(i=0 ; i<len ; i++) {  
    printf("%d ", v[i]);  
}
```



spazio

- Esempio con caratteri

```
char vet[MAX];  
  
...  
for(i=0 ; i<len ; i++) {  
    vet[i] = 'a'+i;  
}  
  
...  
for(i=0 ; i<len ; i++) {  
    printf("%c ", vet[i]);  
}
```

Esempio

- Inizializzare vettore

```
int v[4] = {5, 1, 0, 3};
```

- Stampa

```
for(i = 0; i < 4; i++) {  
    printf("%d ", v[i]);  
}
```

- Stampa dati come se fosse istogramma (orizzontale)

```
for(i = 0; i < 4; i++) {  
    for(j = 0; j < v[i]; j++) {  
        printf("*");  
    }  
    printf("\n");  
}
```


Esempio

- Dichiarazione vettore

```
int v[4];
```

- Inizializza vettore (valori casuali 0 - 20)

```
for(i = 0; i < 4; i++) {  
    v[i] = rand()%21;  
}
```

- Stampa

```
for(i = 0; i < 4; i++) {  
    printf("%d ", v[i]);  
}
```

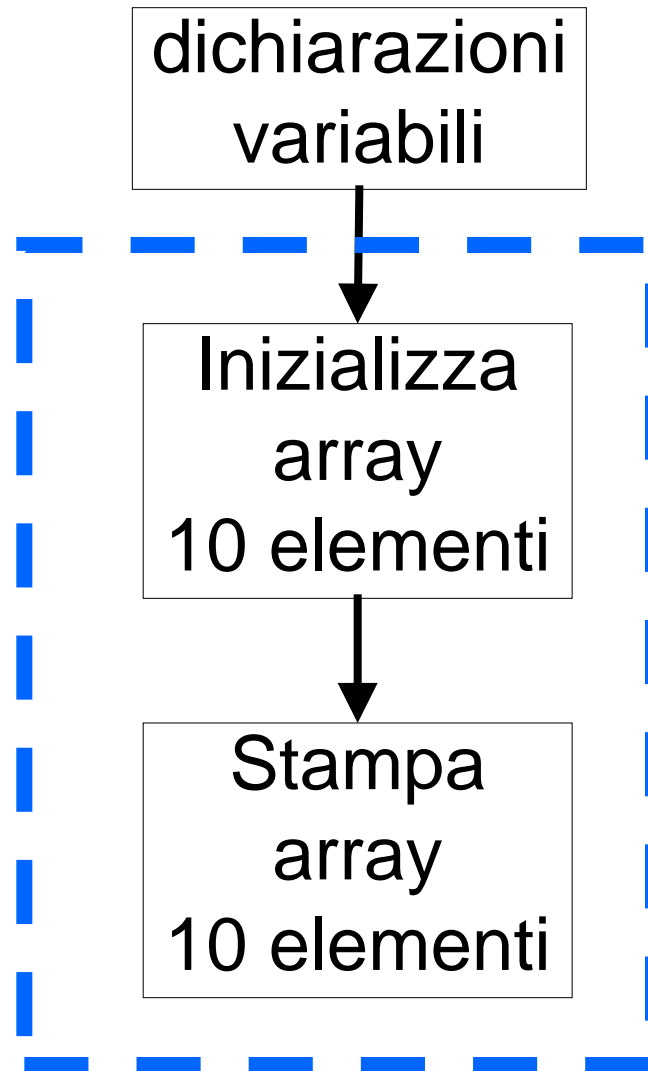
Esercizio 1a

- Scrivere un programma che inizializza un array di 10 elementi di tipo intero con valore pari a 4 e stampa il contenuto dell'array a video.
- Nota
 - Usare costanti ove possibile

Esercizio 1b

- Scrivere un programma che inizializza un array di 10 elementi di tipo intero con **numeri casuali** e stampa il contenuto dell'array a video.
- I numeri casuali devono essere compresi tra 0 e 20.
- Esercizi aggiuntivi
 - Modificare il programma in modo che i numeri casuali siano compresi tra -5 e 20.
 - Usare costanti ove possibile
 - Stampare i valori dell'array solo se sono >10
 - Stampare i valori dell'array in ordine inverso

Esercizio 1 – D.F.



Esercizio 1 – soluzione

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>    // per rand
#define N 10

int main()
{
    int v[N];
    int i;

    i = 0;
    while(i < N){
        v[i] = rand() % (20-0+1)+0; //da 0 a 20
        i++;
    }

    i = 0;
    while(i < N){
        printf("elemento %d = %d\n", i, v[i]);
        i++;
    }

    system("PAUSE");
    return 0;
}
```

Esercizio 1 – soluzione

```
#include <stdio.h>    // per printf
#include <stdlib.h>    // per system
#include <time.h>      // per rand
#define N 10

int main()
{
    int v[N];
    int i;

    for( i=0 ; i<N ; i++){
        v[i] = rand() % (20-0+1)+0; //da 0 a 20
    }

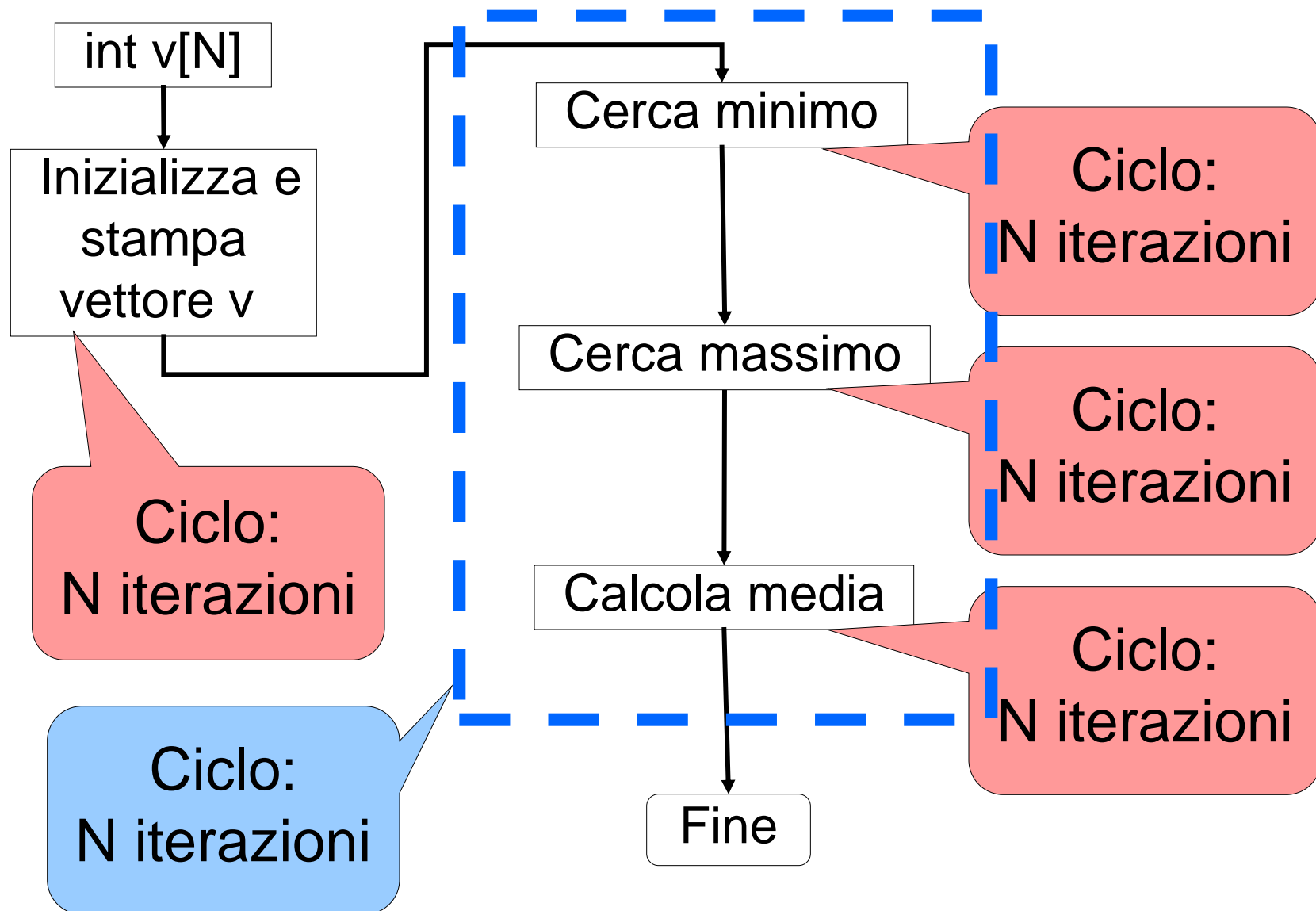
    for( i=0 ; i<N ; i++){
        printf("elemento %d = %d\n", i, v[i]);
    }

    system("PAUSE");
    return 0;
}
```

Esercizio 2

- Scrivere un programma che inizializza un array di 10 elementi di tipo intero con numeri casuali (compresi tra 0 e 20) e li stampa a video.
- Stampare poi:
 - il valore minimo
 - il valore massimo
 - la media

Esercizio 2 – D.F.



Ricerca minimo in array

- Due approcci alla soluzione
- A: tenere traccia del **valore minimo**
 - **v_min** = array[0]
 - Per ogni elemento i di array
 - Se il valore in posizione i è minore di v_min
 - Trovato nuovo minimo
 - **v_min**: contiene valore minimo
- B: tenere traccia delle **posizione** del valore minimo
 - **i_min** = 0
 - Per ogni elemento i di array
 - Se il valore in posizione i è minore di quello in posizione i_min
 - Trovato nuovo minimo
 - **i_min**: contiene indice della posizione del valore minimo
- Logica analoga per la ricerca del massimo in array

valore

posizione

Esercizio 2 – soluzione

```
#define N 10
#define MIN 0
#define MAX 20

int main(){
    int v[N];
    int i, min, max, somma;
    srand(time(0));
    for(i=0 ; i<N ; i++){
        v[i] = (rand() % (MAX - MIN + 1)) + MIN;
        printf("%d ", v[i]);
    }
    printf("\n");
    min = max = v[0]; //min=V[0]; max=v[0];
    somma=0;
    for(i=0 ; i<N ; i++){
        if (v[i] < min)
        { min = v[i];}
        if (v[i] > max)
        { max = v[i]; }
        somma = somma + v[i];
    }
    printf("min: %d\t max: %d\t media: %f\n", min, max,
        (float)somma/N);
    return 0;
}
```

Esercizio 2 – soluzione

```
#define N 10
#define MIN 0
#define MAX 20
int main(){
    int v[N];
    int i, imin, imax, somma;
    srand(time(0));
    for(i=0 ; i<N ; i++){
        v[i] = rand() % (MAX - MIN + 1) + MIN;
        printf("%d ", v[i]);
    }
    printf("\n");
    imin = imax = 0; // imin=0 ; imax=0;
    somma=0;
    for(i=0 ; i<N ; i++){
        if (v[i] < v[imin])
        { imin = i;}
        if (v[i] > v[imax])
        { imax = i; }
        somma = somma + v[i];
    }
    printf("min: %d\t max: %d\t media: %f\n", v[imin], v[imax],
    (float)somma/N);
    return 0;
}
```

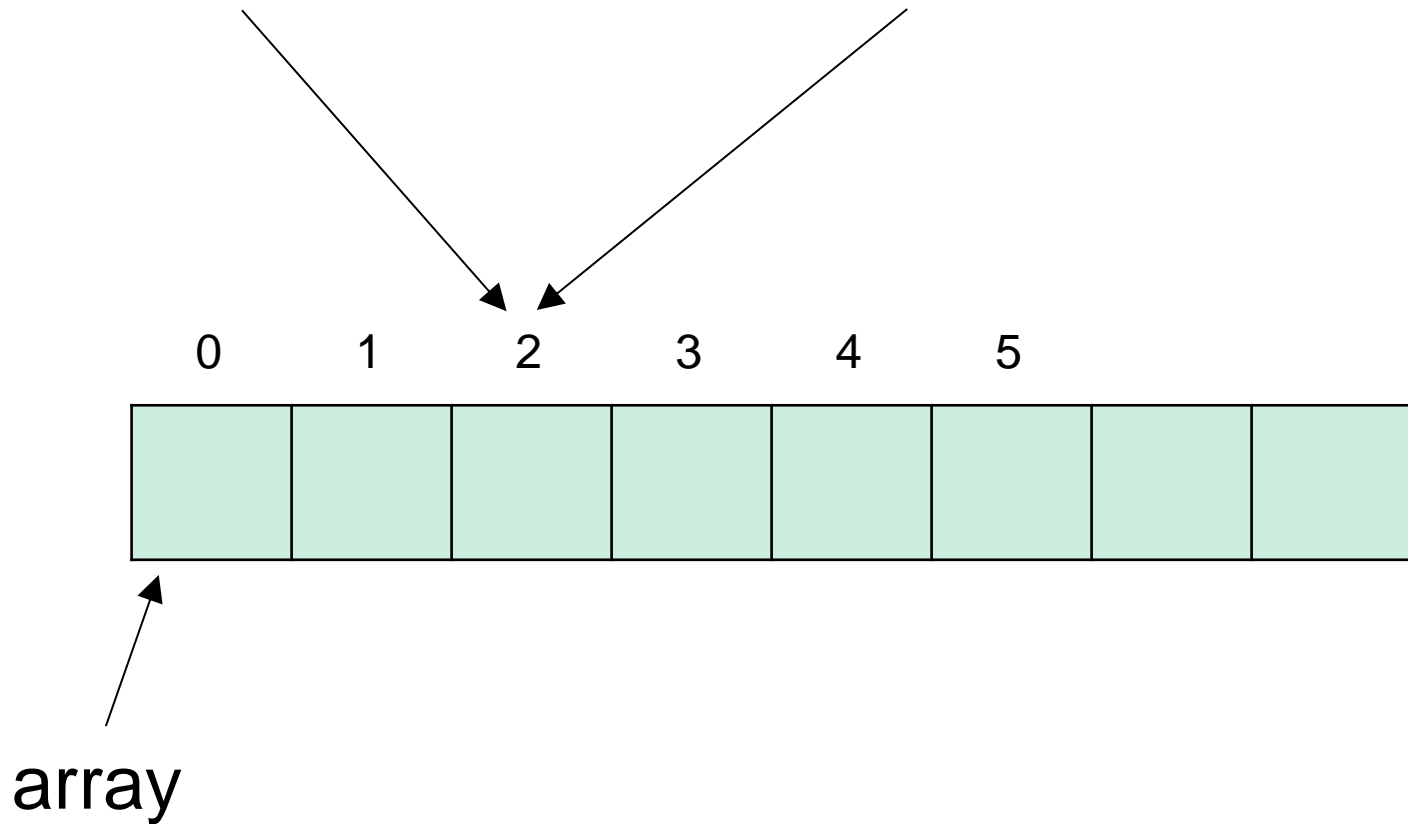
Esercizio 3

- Scrivere un programma che generi casualmente $N=100$ valori tra 0 e `VAL_MAX` e calcolare le occorrenze per ognuno dei valori generati.
- Utilizzare la seguente struttura dati (array) per memorizzare le occorrenze
`occorrenze[VAL_MAX+1]`
 - a) azzerare la struttura dati delle occorrenze
 - b) per ogni valore casual generato, modificare in modo opportuno i valori della struttura dati
 - c) stampare i valori della struttura delle occorrenze
 - d) stampare massima e minima occorrenza

Esercizio 3

indice/posizione

valore generato



Esercizio 3 – soluzione parziale

```
#define N 100
#define VAL_MAX 20
int main() {
    int occ[VAL_MAX+1];
    int i;
    srand(time(0));

    for(i=0 ; i<VAL_MAX+1 ; i++) {
        occ[i] = 0;
    }

    for(i=0 ; i<N ; i++) {
        occ[rand()%VAL_MAX]++;
    }

    for(i=0 ; i<VAL_MAX ; i++) {
        printf("%d: %d\n", i, occ[i]);
    }

    printf("\n");
    return 0;
}
```

Esercizio 4

- Scrivere un programma per calcolare i valori y di un polinomio di grado n

$$y = a_0 + a_1 * x + a_2 * x^2 + \dots + a_n * x^n$$

- Chiedere all'utente il grado del polinomio
 - $0 \leq \text{grado massimo } n \leq \text{MAX}$
 - Controllo input
 - MAX: direttiva di define. Valore 10
- Chiedere all'utente uno alla volta i coefficienti del polinomio e memorizzarli in un array `coeff` (tipo float)
 - Array `coeff`
 - Dimensione massima MAX+1
 - Dimensione effettiva: grado massimo $n + 1$
- Stampare a video i valori y del polinomio calcolato nei punti
 $x = -2.0, -1.5, -1, -0.5, 0, 0.5, 1.0, 1.5, 2.0, 2.5$

Esercizio 4 - Nota

- Costanti da usare
 - MAX: 10 – valore massimo polinomio
- Variabili da usare
 - coeff: array di float – coefficienti del polinomio
 - n: int – grado del polinomio
 - y: float – valore calcolato in un punto x
 - Variabili necessarie per implementare cicli

Esercizio 5

- Estensione esercizio 4
- Cercare il valore massimo calcolato del polinomio (y_{max})
 - Stampare a video il valore massimo del polinomio (y_{max}) e per quale valore di x (x_{max}) è stato calcolato
- Cercare il valore minimo calcolato del polinomio (y_{min})
 - Stampare a video il valore massimo del polinomio (y_{min}) e per quale valore di x (x_{min}) è stato calcolato
- La soluzione NON richiede l'utilizzo di un array per la memorizzazione dei valori y del polinomio calcolati per ogni valore x fornito

Esercizio 5 - Nota

- Costanti da usare
 - MAX: 10 – valore massimo polinomio
- Variabili da usare
 - coeff: array di float – coefficienti del polinomio
 - n: int – grado del polinomio
 - y: float – valore calcolato in un punto x
 - Variabili necessarie per implementare cicli
 - xmax, ymax: float – memorizzare punto di massimo
 - xmin, ymin: float – memorizzare punto di minimo