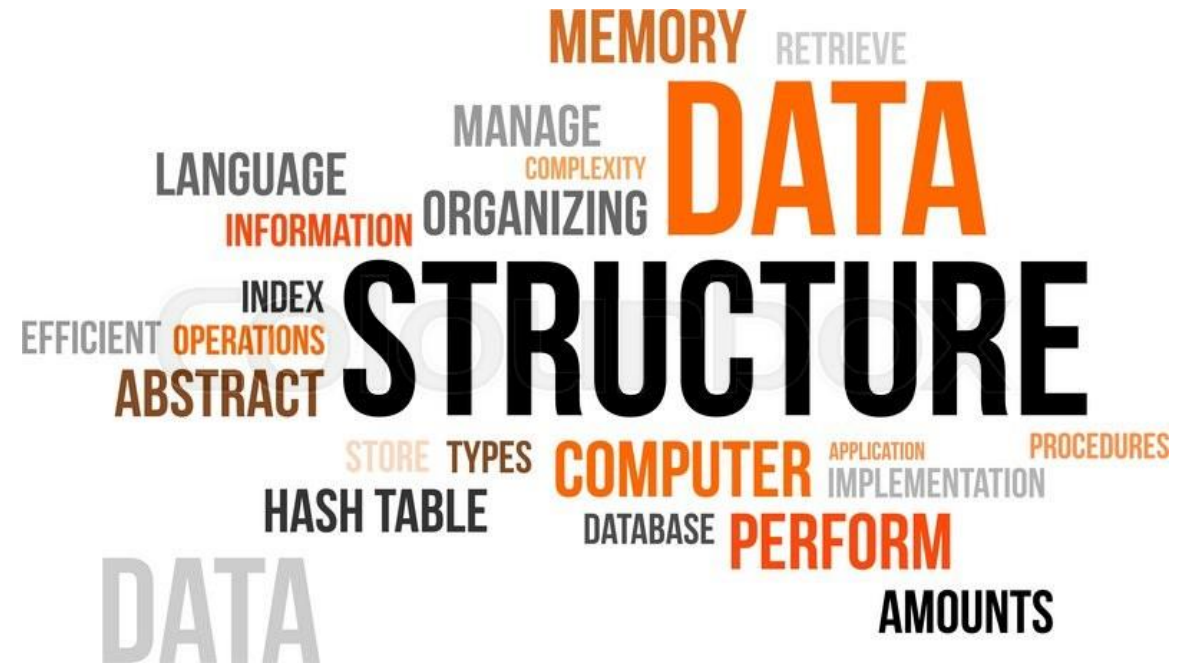


STRUTTURE DATI

Array, vettori e matrici



CHE COSA IMPARERAI A FARE

Definire dati di
tipo enumerativo

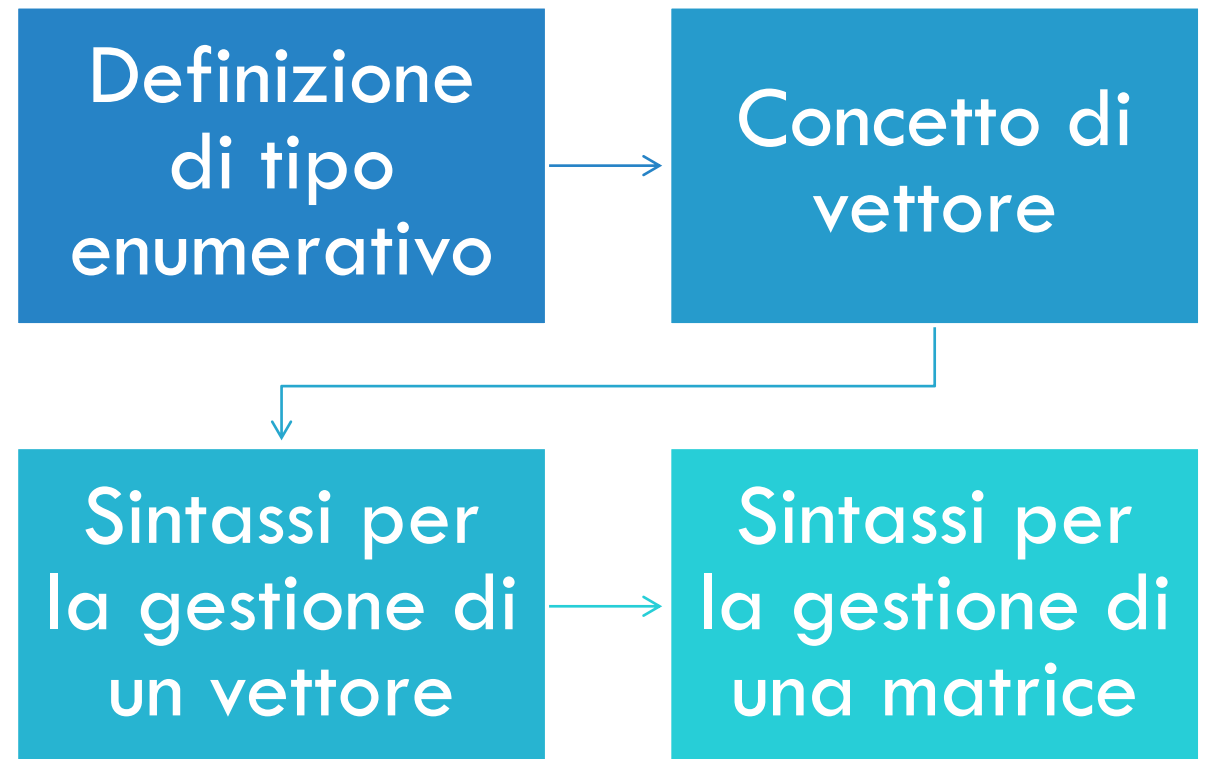
Definire un array
a una dimensione

Caricare un
vettore in
memoria

Definire un
vettore a
dimensione
variabile

Utilizzare una
matrice

CHE COSA DOVRAI STUDIARE





LE ENUMERAZIONI



TIPO ENUMERATIVO

Descrive un insieme finito e ordinato di valori, che possono essere associati a numeri interi positivi

A volte una variabile
può assumere solo
una serie di valori
definiti all'interno di
un insieme

TIPO ENUMERATIVO





ESEMPIO ENUMERATIVO

In un
videogioco, in
quante direzioni
principali ti puoi
spostare?

DEFINIRE ENUMERAZIONE

```
enum nome_enumerazione {elenco_enumerazione};
```


ESEMPIO ENUMERAZIONE

Dichiarazione	<div><div>1</div><div>2</div><div>3</div><div>4</div></div> <code>enum spostamento{avanti, indietro, sinistra, destra};</code>
Inizializzazione	<code>spostamento direzione;</code>
Operazione	<code>direzione = avanti;</code>

```
int main(){  
    enum spostamenti {avanti, indietro,  
sinistra, destra};  
  
    //creo la variabile di enumerazione  
    spostamenti direzione;  
  
    //assegno il valore alla variabile  
    direzione = avanti;
```

```
//decido cosa stampare in base al valore  
dell'enumerazione  
  
    if(direzione == avanti){  
        cout << "Muovi verso Nord di un passo" <<  
endl;  
    } else if(direzione == indietro){  
        cout << "Muovi verso Sud di un passo" <<  
endl;  
    } else if(direzione == sinistra){  
        cout << "Muovi verso Ovest di un passo" <<  
endl;  
    } else if(direzione == destra){  
        cout << "Muovi verso Est di un passo" << endl;  
    }  
}
```



E SE VOLESSIMO
GESTIRE LISTE?

VETTORE

Insieme ordinato
di oggetti
omogenei, ovvero
appartenenti a un
unico tipo



Indice


Dimensione

CARATTERISTICHE VETTORE

TO DO LIST

☒ Too Many To-Do's

☐ How We're Making Lists

☐ Too Much Time 

☐ ? Unknowns & Changes

ESEMPIO LISTA



**TIPO NOMEVETTORE
[DIMENSIONE]**

Vettori in C++

SPECIFICHE VETTORE

01

tipo specifica il tipo di dato comune a tutte le componenti

02

nomevettore è il nome collettivo delle componenti del vettore

03

dimensione è il numero degli elementi contenuti nel vettore

STRING STUDENTI [24];

Lista di 24 persone salvate
come parole (String)

Attenzione!

L'intervallo dei valori possibili per l'indice di un vettore parte da 0 e arriva fino a dimensione-1

[0]

[1]

[2]

[3]

[4]

2

5

1

3

4

RIEMPIRE IL VETTORE

VETTORE

Misure → 15 8 7 16

Indice →	0	1	2	3
----------	---	---	---	---

```
int dim = 4;
```

```
int misure [dim];
```

```
misure[0] = 15;
```

```
misure[1] = 8;
```

```
misure[2] = 7;
```

```
misure[3] = 16;
```

RIEMPIRE IL VETTORE

```
int dim = 4;  
int misure [dim];  
misure[0] = 15;  
misure[1] = 8;  
misure[2] = 7;  
misure[3] = 16;
```

```
int dim = 4;  
int misure [dim];  
for(int i = 0; i<dim; i++){  
    cout<<"Dammi un numero: ";  
    int num;  
    cin>>num;  
    misure[i] =num;  
    cout<<endl;  
}
```

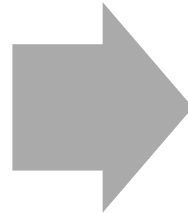
INDEX

Il numero dentro la parentesi quadra serve a specificare la posizione del vettore nella quale io intendo scrivere il valore



ACCESSO AD UNA POSIZIONE DELL'ARRAY

Per accedere ad un elemento specifico dell'array, mi basta usare il nome dell'array seguito dalle parentesi quadre, al cui interno scrivo la posizione a cui mi interessa accedere



Es. Voglio accedere alla posizione 2 di un array lungo dim → `array[2]`

CONSIGLIO

Usate sempre il ciclo FOR per riempire un array o fare operazioni che coinvolgano tutti gli elementi dell'array, dato che sappiamo già la dimensione

OPERAZIONI SINGOLE SUI VETTORI

Stampa

```
cout << "Terzo elemento  
della lista: " << misure[2]  
<< endl;
```

Somma

```
Somma = misure[0] +  
misure[3];
```

ESEMPIO

Vogliamo calcolare il perimetro di un quadrilatero generico

- Inserire i valori dei lati
- Sommare i lati
- Stampare il perimetro

```
int main(){
    int dim = 4;
    int quadrilatero [4];

    //riempiamo il vettore del valore dei lati
    for(int i=0; i<dim; i++){
        cout<<"Scrivi il valore del lato"<<endl;
        cin>>quadrilatero[i];
    }

    //controllo il contenuto del vettore
    for(int i=0; i<dim; i++){
        cout<<"Lato["<<i<<"] = "<<quadrilatero[i]<<endl;
    }

    //sommiamo i lati del quadrilatero
    int perimetro = 0;
    for(int i=0; i<dim; i++){
        perimetro = perimetro + quadrilatero[i];
    }

    //stampo perimetro
    cout << "Il perimetro vale: " << perimetro << endl;
}
```

ALTERNATIVA **ALTERNATIVE** **STRATEGIES** |

```
int [] misure = new int [] {15, 8, 7, 16};
```


[The page contains dense handwritten Japanese text, likely bleed-through from the reverse side. The text is organized into vertical columns and includes various symbols and numbers.]

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

MATRICI



Insieme di dati dello stesso tipo
organizzati in una griglia

- Ogni elemento che compone la matrice è individuato dall'indice di riga e dall'indice di colonna in cui l'elemento è posizionato

	0	1	2	3
0	23	15	3	25
1	3	12	66	78
2	16	21	12	3

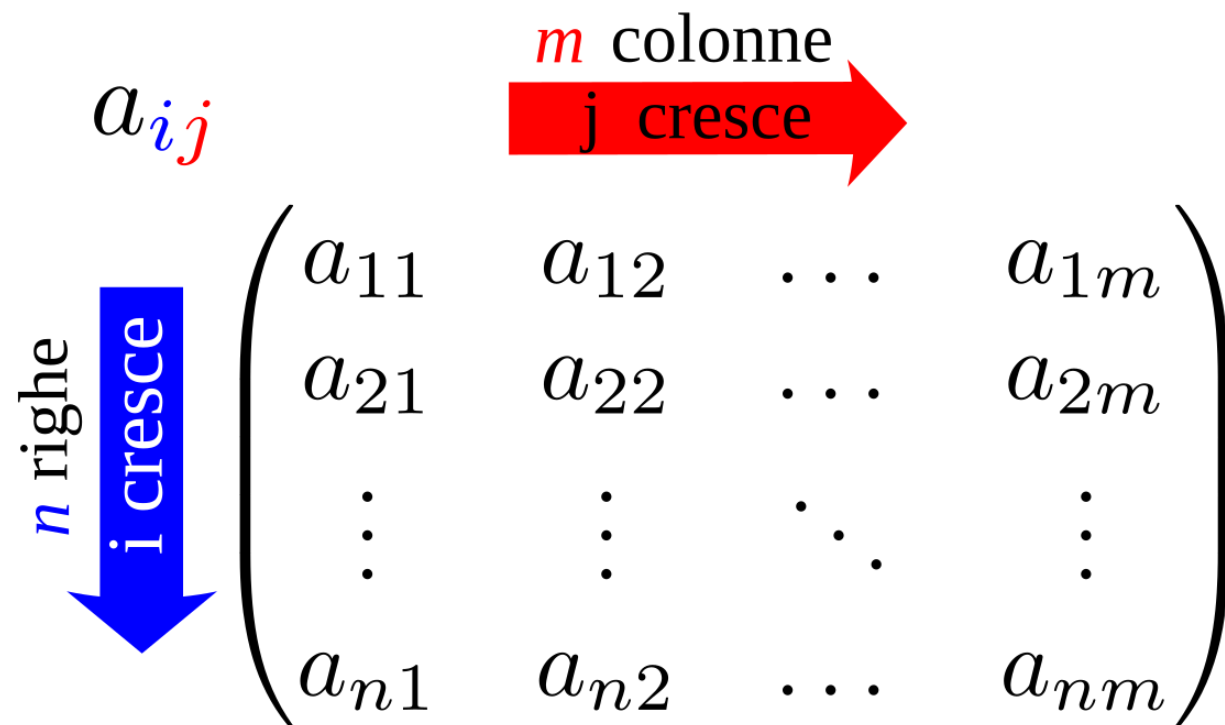


Diagram illustrating a matrix structure with dimensions $n \times m$.


The matrix is represented as:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

Annotations:

- A blue arrow pointing down indicates the row index i increases (i cresce) from 1 to n (n righe).
- A red arrow pointing right indicates the column index j increases (j cresce) from 1 to m (m colonne).

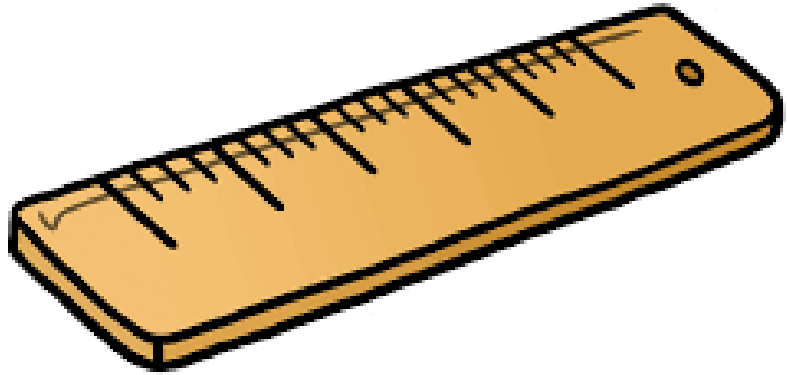
matrice $n \times m$



```
for(int i=0; i<dimensione1;i++){  
    for(int j=0;j<dimensione2;j++){  
        //do something  
    }  
    //do something  
}
```



**POSSO PASSARE UN VETTORE AD
UNA FUNZIONE?**



Funzione(nomeVettore,
dimensioneVettore);

Acquisire da tastiera un vettore di dimensione 5 e stamparlo. Per la realizzazione del programma, utilizzare due funzioni

- `riempiVettore`
- `scriviVettore`

```
void riempiVettore(int v[], int dim){  
    for(int i = 0; i < dim; i++){  
        cout << "Scrivi un numero intero" << endl;  
        cin >> v[i];  
    }  
}
```

```
void scriviVettore(int v[], int dim){  
    cout << "Contenuto del vettore: ";  
    for(int i = 0; i < dim; i++){  
        cout << " " << v[i] << endl;  
    }  
}
```

FUNZIONE 2

```
int main(){  
    int dim = 5;  
    int vettore[5];  
  
    riempiVettore(vettore, dim);  
  
    scriviVettore(vettore, dim);  
}
```




TIME FOR ACTION