

# Esercizi sui vettori

## Esercizio 1 (*Specifica*)

- Scrivere un programma che elimini gli elementi ripetuti in un vettore A, creando un nuovo vettore B senza ripetuti.

### IDEA

- Copio il primo valore di A in B
- Per ogni altro valore di A, prima di inserirlo in B verifico che non sia già presente in B

## Esercizio 1 (Codifica)

```
#include <stdio.h>
#define N 100
void main ()
{
int VALORIA[N], VALORIB[N];
int NB,I,J,NA,TROVATO;

printf("QUANTI VALORI NEL VETTORE A? ");
scanf("%d",&NA);
for (I=0; I<NA; I++)
{printf("introdurre il val. %d:",I+1);
scanf("%d",&VALORIA[I]);    }
```

Vettori ordinati

3

## Esercizio 1 (Codifica)

```
VALORIB[0] = VALORIA[0]; NB = 1;
for (I = 1; I<NA; I++)
{
TROVATO = 0;
for(J=0; (J < NB) && !TROVATO;J++)
if (VALORIA[I]==VALORIB[J])
TROVATO=1;
if(!TROVATO)
{VALORIB[NB]=VALORIA[I];
NB=NB+1;}
}
printf("risultante \n");
for (I=0; I<NB; I++)
{printf("%d\n", VALORIB[I]);}
}
```

Vettori ordinati

4

# Insiemi e Operazioni insiemistiche

- Attraverso i vettori possiamo implementare il concetto di *insieme*
- Su insiemi di valori possiamo implementare
  - Unione
  - Intersezione
  - Differenza

## Esercizio 2 (*Specifica*)

- Scrivere un programma che effettui l'intersezione di due vettori A e B e memorizzi il risultato in un vettore C.

### IDEA

- Scorriamo uno dei due vettori (ad es. A)
- Inseriamo in C ogni elemento di A che è anche elemento di B

## Esercizio 2 (Codifica)

```
#include <stdio.h>
void main ()
{
    int A[100], B[100], C[100];
    int NA,NB,NC,I,J;

    printf("QUANTI VALORI NEL VETTORE A? ");
    scanf("%d",&NA);
    for (I=0; I<NA; I++)
    {printf("introdurre il valore %d:", I+1);
      scanf("%d", &A[I]);    }

    printf("QUANTI VALORI NEL VETTORE B? ");
    scanf("%d",&NB);
    for (I=0; I<NB; I++)
    {printf("introdurre il valore %d:", I+1);
      scanf("%d", &B[I]);    }
```

Vettori ordinati

7

## Esercizio 2 (Codifica)

```
NC = 0; /* NC: dimensione del nuovo vettore */
for(I=0; I < NA; I++)
    for(J=0; J< NB; J++)
        if (A[I] == B[J])
            { C[NC]= B[J];
              NC= NC+1;}
for (I=0; I<NC; I++)
    {printf("%d\n", C[I]);}
}
```

Vettori ordinati

8

## Esercizio 3

- Scrivere un programma che dati due vettori di al più N elementi di tipo intero, presenti all'utente il seguente menù per la gestione di vettori:

- 1 – creare il vettore unione
- 2 – creare il vettore intersezione
- 3 – creare il vettore differenza
- 4 – fine programma

Il programma deve consentire all'utente di eseguire una o più operazioni fino a quando non digita l'opzione di uscita (4).

Al termine di ogni operazione eseguita il programma deve stampare il contenuto del vettore ottenuto.

## Vettori e Spazi vettoriali

- Attraverso i vettori possiamo implementare *vettori in uno spazio vettoriale*
- I vettori possono essere *sommati e moltiplicati* per uno scalare usando le operazioni che definiscono lo spazio vettoriale a cui appartengono

$$\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} v_1 + w_1 \\ \vdots \\ v_n + w_n \end{pmatrix}$$

$$\lambda \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \lambda v_1 \\ \vdots \\ \lambda v_n \end{pmatrix}$$

- La somma di due vettori in  $K^n$  è definita come la somma componente per componente
- Il prodotto di un vettore per uno scalare è definito come il prodotto di ogni componente per lo scalare

# Norma, prodotto scalare

- Il prodotto scalare euclideo di due vettori  $v$  e  $w$  in  $K^n$  è il numero:
$$\langle v, w \rangle = v_1 w_1 + v_2 w_2 + v_3 w_3 + \dots + v_n w_n$$
- La norma euclidea (modulo) di un vettore  $v$  in  $K^n$  è il numero:
$$\|v\| = \sqrt{v_1^2 + v_2^2 + v_3^2 + \dots + v_n^2}$$
- La norma euclidea può essere scritta come:
$$\sqrt{\langle v, v \rangle} = \sqrt{v_1 v_1 + v_2 v_2 + v_3 v_3 + \dots + v_n v_n}$$

Esercizi sui vettori

## Prodotto scalare

```
#include <stdio.h>
main() {
    int i, dim;
    float modulo1, modulo2;
    float vett1[100], vett2[100];

    . . .

    //prodotto scalare dei due vettori
    prodotto_scalare=0;
    for(i=0; i<dim; i++)
        prodotto_scalare+=(vett1[i]*vett2[i]);

    printf("\nProdotto scalare: %f\n\n", prodotto_scalare);
}
```

Esercizi sui vettori

# Norma

```
#include <stdio.h>
#include <math.h>
main() {
    int i,dim;
    float modulol;
    float vett1[100];

    printf("Dammi la dimensione del vettore: ");
    scanf("%d",&dim);

    //lettura di un vettore di reali
    for(i=0;i<dim;i++){
        printf("Primo vettore - componente %d: ",i);
        scanf("%f",&vett1[i]);
    }
    modulol=0;
    for(i=0;i<dim;i++){
        modulol+=(vett1[i]*vett1[i]);
    }
    modulol=sqrt(modulol);

    printf("\nPrimo vettore - modulo: %f\n\n",modulol);
}
```

Esercizi sui vettori

## Angolo fra due vettori

- Il prodotto scalare fra vettori  $v$  e  $w$  in  $K^n$  è definito anche come:

$$\langle v, w \rangle = \|v\| \cdot \|w\| \cos(\theta)$$

con  $\theta$  angolo fra i due vettori

- L'angolo fra due vettori quindi si può ricavare dalla relazione:

$$\theta = \arccos\left(\frac{\langle v, w \rangle}{\|v\| \cdot \|w\|}\right)$$

il valore di  $\theta$  è espresso in radianti

$$\alpha_{(\text{gradi})} = \theta_{(\text{radianti})} * 360/2\pi$$

Esercizi sui vettori

# Angolo fra due vettori

```
#include <stdio.h>
#include <math.h>
main() {
    int i,dim;
    float modulo1, modulo2, ang;
    float vett1[100], vett2[100];
    . . .
    modulo1=modulo2=prodotto_scalare=0;
    for (i=0; i<dim; i++) {
        modulo1+=(vett1[i]*vett1[i]);
        modulo2+=(vett2[i]*vett2[i]);
        prodotto_scalare+=vett1[i]*vett2[i];
    }
    modulo1=sqrt(modulo1); modulo2=sqrt(modulo2);

    //angolo fra vettori
    ang=acos(prodotto_scalare/(modulo1*modulo2))*180.0/PIGRECO;
    printf("Angolo fra i due vettori: %f gradi\n",ang);
}
```

Esercizi sui vettori

## Esercizio 4

- Scrivere un programma che dati due vettori di al più N elementi di tipo intero, presenti all'utente il seguente menù per la gestione di vettori:

- 1 – stampare il vettore somma
- 2 – stampare il vettore differenza
- 3 – calcolare il modulo dei due vettori
- 4 – calcolare il prodotto scalare dei due vettori
- 5 – calcolare l'angolo fra i due vettori
- 6 – fine programma

Il programma deve consentire all'utente di eseguire una o più operazioni fino a quando non digita l'opzione di uscita (6). Al termine di ogni operazione eseguita il programma deve stampare il contenuto del vettore ottenuto.