

Array

Contenuti rieditati delle slide della
Prof. L. Caponetti

Dati strutturati

- I dati **strutturati** sono aggregati di dati **elementari**:
 - Vettori e matrici in matematica
 - Elenchi telefonici
 - Dizionari
 - Code, liste, pile
 -

Tipi strutturati

- I tipi **strutturati** possono essere costruiti mediante tipi già definiti (costituenti)
- Gli elementi del dominio di un tipo di dato strutturato sono ottenuti aggregando gli elementi, **componenti**, appartenenti ai domini dei tipi costituenti

Strutture dati

- Utilizziamo la seguente notazione:
 - Indichiamo con **struttura dati** un tipo di dato il cui dominio è composito, cioè composto da dati più elementari
 - Indichiamo con **tipo** di dato un tipo di dato il cui dominio è elementare cioè costituito da elementi atomici non decomponibili (es. int)

Strutture dati

- Una struttura dati è costituita da un insieme di componenti su cui sono definiti un insieme di operatori per:
 - **Selezionare una componente**
 - **Cancella una componente**
 - **Aggiungere una componente**
 -

Strutture dati - linearità

- Le strutture dati possono essere **lineari**
 - Una struttura dati lineare può essere vista come una sequenza di dati in cui è possibile individuare la 1° componente, la 2° e così via fino ad esaurire la lunghezza della sequenza, ad esempio i vettori
- Le strutture dati possono essere **non lineari**, ad esempio la matrice

Array

- L'**array** è una struttura dati omogenea, cioè costituita da componenti dello stesso tipo, detto **tipo base**
- L'array è una struttura ad accesso **casuale**, tutte le componenti possono essere selezionate a caso, nello stesso tempo, mediante un indice

Array

- Una variabile array è individuata da un **unico identificatore**
- Ogni componente è selezionata mediante un indice
- Array monodimensionale: **un array ad una dimensione è costituito da un insieme finito di elementi omogenei, in corrispondenza biunivoca con un insieme di indici**

Array - dichiarazione

- Nei linguaggi di alto livello è necessario dichiarare:
 - Il tipo delle componenti o tipo base (può essere semplice o strutturato)
 - Il numero delle componenti (costante o variabile)
 - Il tipo dell'indice
- **Nel linguaggio C l'indice deve essere un intero e variare nel dominio $[0, n-1]$ se n è il numero delle componenti**

Esempio in C

```
#define DIM 100 // numero di elementi del vettore  
int a[DIM]; // dichiarazione del vettore  
int i; // dichiarazione dell'indice
```

- In C e in Pascal non esiste il tipo array, ma la possibilità di costruirlo aggregando componenti omogenee
- Gli **operatori** che si possono applicare sono soltanto quelli relativi al **tipo delle componenti**

Array – selezione di una componente

- Sia x un array di n elementi
 - **Selezione**: la selezione di una componente è possibile mediante il nome della variabile array ed un valore dell'indice
 - $x[i]$ seleziona la componente di indice i** (i varia tra 0 ed $n-1$)

Array – aggiornamento di una componente

- Sia x un array di n elementi
 - **Aggiornamento:** l'aggiornamento del valore della componente di indice i si può ottenere:
 - **Mediante una istruzione di assegnazione:** $x[i] = v$
 - **Mediante la lettura di un valore:** `scanf(«%d», &x[i])`

Rappresentazione in memoria centrale

- Una variabile array è rappresentata in memoria centrale in modo **sequenziale**, cioè le diverse componenti sono organizzate in locazioni di memoria contigue



Rappresentazione in memoria centrale

- Sia x un array di n componenti (con n costante), e sia i l'indice variabile da 0 a $n-1$
- Le componenti di x sono rappresentate in memoria a partire **dall'indirizzo iniziale di x** , $\text{ind}(x)$, uguale all'indirizzo della prima componente



Rappresentazione in memoria centrale



- Sia L l'occupazione di memoria in byte di una componente di x , dipendente dal tipo base
 - L'occupazione di memoria della variabile x è data da $n * L$
 - L'indirizzo di un generico elemento i è dato da
$$\text{ind}(x[i]) = \text{ind}(x) + i * L$$

Strutture dati - omogeneità

- Un linguaggio di programmazione offre diversi metodi di strutturazione dei dati:
 - Si possono aggregare dati dello **stesso tipo**, ad esempio in un array
 - Oppure di **tipo diverso**, ad esempio in un record

Strutture dati - accesso

- Si possono avere diversi **metodi di accesso** alle diverse componenti
 - **Accesso diretto ad ogni componente**
 - **Accesso sequenziale ad ogni componente**
 - **Accesso diretto solo ad alcune componenti**

Strutture dati - dimensione

- Strutture dati a **dimensione fissa** – il numero delle componenti non cambia
- Strutture dati a **dimensione variabile** - il numero delle componenti è variabile