

Dati strutturati

Problema

Leggere da tastiera 20 numeri interi e ristamparli in ordine inverso.

Sembra semplice... ma...

Come si fa???

Idea 1

- Definisco tante variabili quanti sono i numeri da inserire
- Leggo e memorizzo i valori nelle variabili
- Le rileggo al contrario

Sembra facile...

Problema

Quando provo a tradurre la mia idea in C mi accorgo di un problema. Dovrei fare:

Per 20 volte:

- leggi un numero

- metti il primo nella prima variabile

- il secondo nella seconda

- il terzo nella terza...

Oppure

Per i che va da 1 a 20
leggi un numero
metti il numero nella i -esima variabile

Ma anche qui ho bisogno di 20 variabili, non riesco ad usare il ciclo perché nell'istruzione devo cambiare sempre il nome della variabile.

Mi tocca scrivere le istruzioni 20 volte???

Ragioniamo

La situazione precedente si può riassumere così:

- Devo leggere tanti dati in ordine
- I dati sono tutti dello stesso tipo

La soluzione

Gli array o vettori, ovvero un particolare tipo di dato strutturato.

Un array è una disposizione ordinata di elementi omogenei tra loro.

Vettori o array a una dimensione

Nei programmi realizzati finora abbiamo utilizzato variabili di tipo primitivo in grado di contenere un solo valore per volta, ora vedremo come sia possibile utilizzare variabili contenenti più valori dello stesso tipo.

A cosa servono?

Esistono situazioni in cui le variabili «semplici» non sono sufficienti alle esigenze del programmatore; ad esempio: dati gli N voti di una verifica, voglio sapere quali (o quanti) voti sono superiori al valore medio. Usando una sola variabile `VOTO`, il cui input avviene in un ciclo che si ripete N volte e in cui sommo i valori gestendo un accumulatore, al termine del ciclo posso calcolare il valore medio, ma ho perso i voti inseriti (dovrei reinserirli!?!).

Variabili complesse

Abbiamo bisogno di una variabile in grado di memorizzare (temporaneamente, finché dura l'esecuzione del programma) i dati per poterli riutilizzare nel corso del processo (esecuzione del programma).

Array o vettore

Un array consente di memorizzare temporaneamente più **dati dello stesso tipo** (primitivi o di riferimento) utilizzando una sola variabile; è un dato strutturato, i cui elementi (o componenti) si distinguono tra loro per la posizione che occupano all'interno, per l'indice.

Esempio

Variabile voti, contiene un elenco di numeri interi.

Voti

6	6	6	7	7	7	7	9	4	7
0	1	2	3	4	5	6	7	8	9

Nota bene:

- Il vettore contiene 10 elementi numerati da 0 a 9
- Tutti gli elementi contenuti sono dello stesso tipo: interi

Generalizzando

	indice	elemento
Primo elemento	0 ↔	Elemento[0]
Secondo elemento	1 ↔	Elemento[1]
...	2	Elemento[2]
....	3	Elemento[3]

n-esimo elemento	N-1	Elemento[N-1]

L'indice

L'utilizzo dell'indice determina un ordine interno conferendo alla variabile una struttura, cioè un'organizzazione interna che facilita il reperimento e il trattamento dei dati nelle elaborazioni.

Definizione

Quindi l'array è un insieme ordinato di dati **dello stesso tipo**, in cui esiste una **corrispondenza biunivoca tra indice ed elemento.**

Come si usa

Questo tipo di variabili si usano quasi sempre insieme ai cicli.

Esempio dei voti:

- in una prima fase carico tutti i dati necessari, i voti, facendo un ciclo su tutti gli elementi del vettore
- poi scandisco, o “leggo” con un ciclo dal primo all’ultimo elemento (da 0 a N-1) l’array sommando i voti (anche se potevo farlo durante l’inserimento) per calcolare la media

Continua soluzione es. voti

- successivamente rileggo l'array controllando quali (o contando) voti superano il valore medio.

Importante

Nel linguaggio C (come in molti altri) la memorizzazione di N elementi in un array parte dalla posizione 0: la prima posizione è 0, la seconda è 1 ...l'ultima posizione è la $N-1$; quindi, se devo memorizzare N elementi, le posizioni vanno da 0 a $N-1$.

Creazione e dichiarazione

Un array si dichiara con:

tipo nomevettore[dim];

ad esempio float VOTO[26]

oppure int VOTO[MAX], con MAX costante.

Occorre definire a priori una dimensione fisica massima (26, oppure anche usando una costante MAX); successivamente è possibile utilizzare anche solo una parte degli elementi.

Caricamento dei dati

Devono essere acquisiti i valori da memorizzare in ogni cella dell'array; ogni **elemento** è identificato dalla sua posizione che va specificata tra parentesi quadrate

(es. VOTO[1], VOTO[2],...VOTO[9])

Elaborazione

Si scandisce con un ciclo di lettura l'array (considerando la dimensione logica, ovvero gli elementi riempiti), effettuando le elaborazioni richieste (visualizzazione degli elementi, calcolo della somma degli elementi, della media, ricerca di massimo/minimo, ricerca della posizione di un elemento noto il valore, ricorrenza di un elemento, ordinamento crescente/decescente degli elementi...)

16_array_senza_funzioni.c

Analisi del codice