

I VETTORI

“Dati N numeri, calcolare la percentuale di ogni numero sul totale”. Cerchiamo di svolgere questo esercizio per comprendere, se ce ne sono, le problematiche che esso nasconde. Procediamo con l’esecuzione della prima parte:

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

int main()
{
    int i, N, num, totale=0;

    do
    {
        printf("Inserire la dimensione del vettore: ");
        scanf("%d", &N);

    }
    while(N<1 || N>100);
    for(i=0; i<N; i++)
    {
        printf("Inserire l'elemento ");
        scanf("%d", &num);

        totale=totale+num;
    }
}
```

A questo punto si dovrebbe calcolare la percentuale di ogni numero sul totale, ma è impossibile effettuare questo calcolo, poiché avremmo bisogno di tutti i numeri inseriti dall’utente, ma ormai persi in quanto ad ogni nuova lettura è stato salvato nella variabile “num” il nuovo valore inserito perdendo il precedente. Cosa contiene la variabile “num” alla fine del ciclo enumerativo di lettura dei numeri? In essa è presente l’ultimo valore inserito dall’utente, pertanto, solo per quest’ultimo, possiamo calcolare la sua percentuale sul totale.

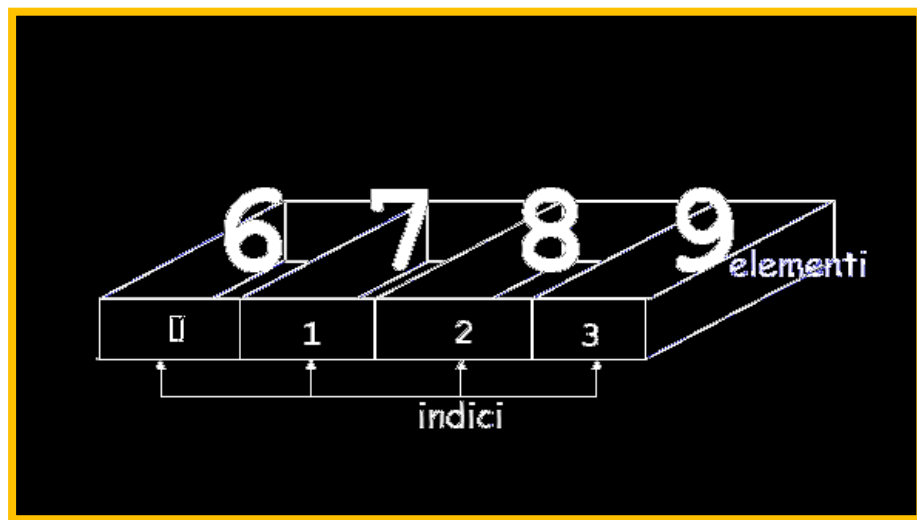
Se N fosse un numero abbastanza piccolo, ad esempio uguale a 3, si potrebbe pensare di usare tre variabili diverse dello stesso tipo per contenere tutti i numeri; se N fosse uguale per esempio a 100, è chiaro che sarebbe impossibile gestire 100

variabili dello stesso tipo con nomi diversi: invece di utilizzare tante variabili diverse dello stesso tipo, in programmazione si usa l'array, ossia una variabile che si definisce strutturata, in cui è possibile memorizzare più valori dello stesso tipo.

Gli array possono essere multidimensionali, argomento che si affronterà il prossimo anno, e monodimensionali; in questo secondo caso prendono anche il nome di vettori, argomento della presente lezione.

Per comprendere il significato del vettore portiamo un esempio tratto dalla quotidianità.

Disegniamo una scatola suddivisa in tanti scomparti quanti sono i dati da memorizzare. La scatola è il *vettore*, ogni scomparto rappresenta l'*elemento*; esso contiene un unico dato ed è individuato da un numero progressivo detto *indice* che specifica la posizione dello scomparto e quindi dell'elemento all'interno della scatola, ovvero all'interno del vettore:

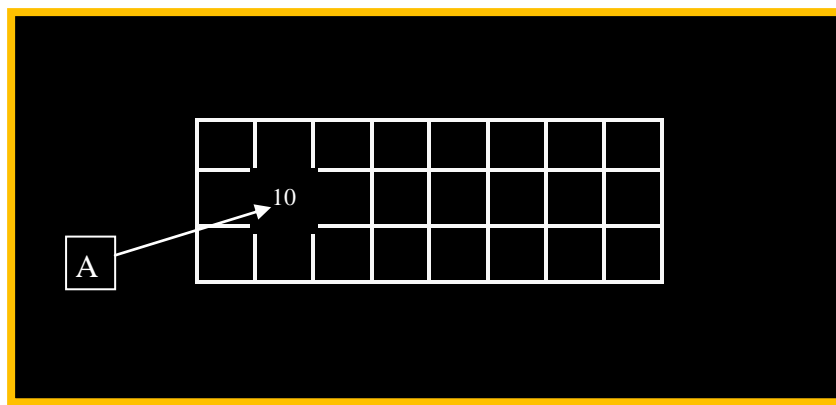


L'insieme degli elementi viene cioè organizzato in modo che ci si possa riferire ad esso come ad una singola entità, attraverso l'identificatore del vettore (scatola) ed allo stesso tempo è possibile accedere ad una qualsiasi delle sue componenti indicandola con un indice che corrisponde alla sua posizione nel vettore. Per cui ciascuna componente del vettore viene indicata dal nome del

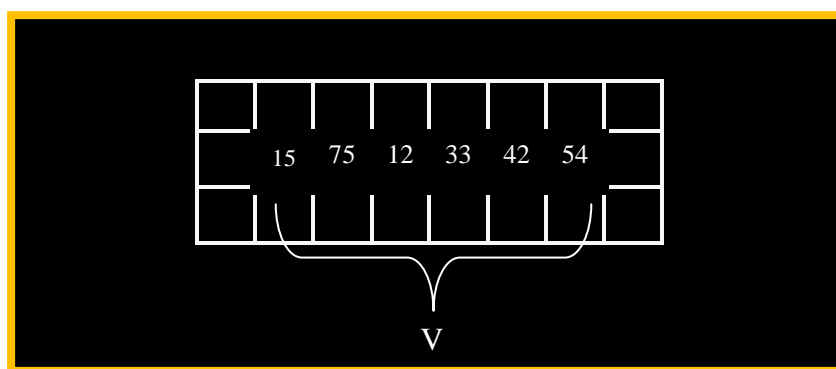
vettore seguito dall'indice racchiuso tra parentesi quadre. Così relativamente al vettore *scatola* il valore di *scatola*[0] corrisponde al 6, quello di *scatola*[1] al 7 e così via.

Ricollegandoci a quanto detto in precedenza, un insieme di 100 numeri, può essere indicato con 100 variabili A, B, C, ..., oppure semplicemente con V[i] con $i=0, \dots, 99$, dove V[0] è il primo numero, V[1] è il secondo e così via.

Ma cosa accade in memoria quando si manipolano i vettori? Disegniamo alla lavagna la situazione della memoria quando si utilizza una variabile elementare come quelle usate fino ad ora. Considerata una variabile A di tipo intero in memoria viene riservata una cella di memoria in grado di contenere un elemento di tipo intero:



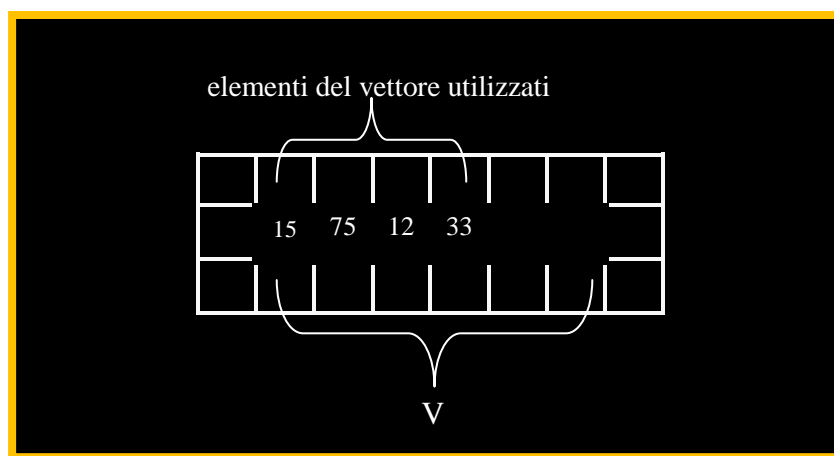
Supponiamo invece di rappresentare le età di 6 persone, si potrà utilizzare un vettore formato da 6 elementi interi chiamato V. In questo caso alla variabile V viene associata un'area di memoria formata da 6 *celle contigue* ciascuna delle quali è in grado di contenere un elemento di tipo intero:



Poiché quando si utilizza il vettore in un esercizio non si conosce a priori il numero effettivo di elementi che l'utente inserirà, all'inizio dell'esercizio si definisce un numero arbitrario abbastanza grande che rappresenta il massimo numero di elementi che il vettore potrà contenere, che prende il nome di dimensione del vettore.

Nel corso dell'esercizio, dopo aver chiesto all'utente il numero di elementi che realmente intende inserire nel vettore, verrà utilizzata soltanto una sua parte, ovvero quella utile a contenere il numero di valori a cui l'utente è interessato. Supponiamo, ad esempio, che l'utente voglia inserire solo le età di 4 persone nel vettore V.

Quanto esposto è stato rappresentato graficamente in questo modo:



A meno che non si evinca già dalla traccia la dimensione del vettore e in tal caso si può dichiarare un vettore della dimensione necessaria sin dall'inizio dell'esercizio.

Se, dunque, la variabile che stiamo utilizzando è un vettore, essa deve essere sempre usata all'interno dell'algoritmo accompagnando il nome della variabile con un indice.

In questo modo, incrementando opportunamente l'indice, è possibile caricare, scorrere e stampare facilmente gli elementi di un vettore. Infatti l'usuale costrutto per effettuare tali operazioni è il ciclo di iterazione enumerativa.

In particolare, per il caricamento di un vettore, operazione che consente di assegnare valori agli elementi di un vettore, e per la stampa dello stesso, si possono eseguire i passi effettuati nel seguente esempio:

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

int main()
{
    int V[100], i, N;

    do
    {
        printf("Inserire la dimensione del vettore: ");
        scanf("%d", &N);
    }
    while(N<1 || N>100);
    for(i=0; i<N; i++)
    {
        printf("Inserire l'elemento V[%d] del vettore: ", i+1);
        scanf("%d", &V[i]);
    }
    printf("Gli elementi del vettore sono: \n");
    for(i=0; i<N; i++)
    {
        printf("%d\n", V[i]);
    }
    system("PAUSE");
}
```

Acquisiamo la dimensione effettiva del vettore

Carichiamo gli elementi del vettore: il contatore indica il numero di posto all'interno del vettore in cui deve essere inserito il dato digitato, che varia da 0 ad N-1, laddove N è la dimensione effettiva del vettore. Con l'uso dei vettori i dati inseriti da tastiera vengono immagazzinati in memoria in modo organizzato e rimangono disponibili per le successive elaborazioni.

Stampiamo gli elementi del vettore: il contatore indica la posizione di ciascun elemento del vettore.

Abbiamo tutti gli strumenti per risolvere l'esercizio proposto all'inizio della lezione:

```

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

int main()
{
    int V[100], i, N, totale=0;

    do
    {
        printf("Inserire la dimensione del vettore: ");
        scanf("%d", &N);
    }
    while(N<1 || N>100);
    for(i=0; i<N; i++)
    {
        printf("Inserire l'elemento V[%d] del vettore: ", i+1);
        scanf("%d", &V[i]);
        totale=totale+V[i];
    }
    printf("Le percentuali di ogni valore inserito sul totale sono le seguenti: \n");
    for(i=0; i<N; i++)
    {
        printf("%f\n", (float)V[i]/totale*100);
    }
    system("PAUSE");
}

```

Esercizio da svolgere a casa: “Acquisire da tastiera un vettore e stampare il contenuto e la posizione degli elementi positivi in esso presenti”.