

Laboratorio 5: Array

Sara Montagna

17 Aprile 2020

1 PRIMI ESERCIZI CON GLI ARRAY

ESERCIZIO 1 Scrivere un programma per la stampa di un vettore dall'ultimo al primo elemento.

ESERCIZIO 2 Scrivere un programma per inizializzare un vettore con input da tastiera. Il programma deve prima di tutto chiedere all'utente di inserire la dimensione del vettore, e poi di inserire gli elementi. Nota: questo programma richiede un compilatore conforme allo standard C99 (quello incluso in Code::Blocks lo e'), in quanto usa delle caratteristiche non disponibili in ANSI C (es., variable length arrays, dichiarazione di variabili in mezzo al codice, etc.)

ESERCIZIO 3 Scrivere un programma che, dato un vettore di interi v di 10 elementi il cui valore, compreso tra 0 e 99, viene inizializzato in modo random attraverso la funzione `int rand(void)`, trovi il massimo tra gli elementi di v e ne stampi il valore a video. Ricordiamo che l'espressione `rand() % 100` restituisce un intero "casuale" nell'insieme $0, \dots, 99$.

2 ESERCIZI AVANZATI CON GLI ARRAY

ESERCIZIO 4 Implementare all'interno della funzione `main()` un algoritmo che, a partire dai due array `v1` e `v2` di `double` e nota la loro lunghezza `n` calcoli il loro prodotto scalare e lo stampi a video. Ricordiamo che il prodotto scalare è definito come:

$$\sum_{i=0}^{n-1} v1[i] \times v2[i]$$

NOTA: la lunghezza degli array deve essere la stessa per entrambi, fissa e definita prima della dichiarazione dell'array o come costante o attraverso la direttiva `#define`

ESERCIZIO 5 Scrivere un programma che, dati 6 cassette di resistenze etichettati con i seguenti valori (in Ohm): 10, 15, 22, 33, 47, 68, stabilisca dove riporre una nuova resistenza di valore misurato R .

- Si assuma che i valori assegnati a ciascuna cassetto abbiano una tolleranza di +/- 10% ovvero, il cassetto etichettato 15 possa contenere resistenze con $15 \times 0.9 \leq R \leq 15 \times 1.1$
- Nota: gli intervalli NON si sovrappongono: è possibile che R sia tale da non poter essere inserito in alcun cassetto; in tal caso il programma lo deve segnalare con un opportuno messaggio d'errore
- Assumere che tutti i valori siano di tipo `float` o `double`

ESERCIZIO 6 Scrivere un programma C che, dato un array composto da 40 valori interi tutti compresi tra 0 e 9 (estremi inclusi), stampi le frequenze (numero di occorrenze) di ciascuno dei 10 possibili valori. Ad esempio, dato:

```
#define N 40
...
int valori[N] = {0, 1, 1, 0, 0, 2, 4, 6, 5, 5,
                7, 2, 3, 8, 9, 9, 1, 3, 2, 4,
                3, 2, 4, 2, 3, 2, 6, 5, 4, 7,
                7, 8, 7, 2, 3, 0, 9, 9, 1, 3};
```

il programma deve stampare qualcosa come:

```
0 : compare 4 volte
1 : compare 4 volte
2 : compare 7 volte
3 : compare 6 volte
4 : compare 4 volte
5 : compare 3 volte
6 : compare 2 volte
7 : compare 4 volte
8 : compare 2 volte
9 : compare 4 volte
```

Si può implementare la soluzione all'interno del metodo `main()` senza l'ausilio di funzioni d'appoggio.

Facoltativo: Stampare le frequenze in modo "pseudo-grafico", usando ad esempio degli asterischi come segue:

```
0 : ****
1 : ****
2 : *****
3 : *****
4 : ****
5 : ***
```

6 : **
7 : ****
8 : **
9 : ****

ESERCIZIO 7 Progettare e implementare un algoritmo che dato un array a di interi avente lunghezza n maggiore di zero, stabilisce e stampa a video se l'array è ordinato in senso non decrescente. In altre parole, l'algoritmo deve stampare "crescente" solo se si ha $a[0] \leq a[1] \leq \dots \leq a[n - 1]$.

ESERCIZIO 8 Completare il programma `es_sinh.c` presente in allegato all'esercitazione (NB: questo esercizio è tratto dalla prova d'esame del 16/9/2019).