

Algoritmi elementari

- Somma di N numeri: la faccio ponendo inizialmente uguale a zero la variabile che deve contenere la somma, e aggiungendo un termine alla volta. Non bisogna dimenticare di azzerare inizialmente la somma
- Prodotto di N numeri: lo stesso problema di prima, solo che la variabile che contiene il prodotto deve essere posta inizialmente uguale a uno.
- Scambio di due numeri: non si possono scambiare a e b semplicemente scrivendo $a=b$; e poi $b=a$; perché così alla fine a e b conterrebbero lo stesso valore: ci vuole una variabile intermedia in cui salvare il valore iniziale di a.

- Divisioni tra interi: per un computer $1/2=0$ e $1./2.=0.5$. occorre fare attenzione alla divisione tra interi.
- Array: in C, tutti i vettori partono da zero, quindi $a[100]$ ha componenti $a[0], a[1], \dots, a[99]$. Cercare di accedere ad $a[100]$ è un errore che porta a risultati imprevedibili.
- I calcoli numerici con numeri reali sono sempre affetti da errori di troncamento, quindi bisogna evitare condizioni del tipo: se $x==y$ allora... nelle istruzioni condizionali.

- Allocazione dinamica: se mi serve un array (=vettore) `double v` di 100 elementi posso riservare la memoria dall'inizio con la dichiarazione `double v[100]` oppure allocare dinamicamente usando, per esempio, la funzione *malloc* (in questo caso devo includere *stdlib.h*). *malloc* restituisce un puntatore void e ha come argomento il numero di byte da allocare (in questo caso $100 * \text{sizeof}(\text{double})$). Il puntatore a void può essere convertito in un puntatore a qualunque altro tipo. La memoria allocata deve essere liberata, quando non serve più, con l'istruzione *free(v)*.