**Attività 2 – Operazioni su array**

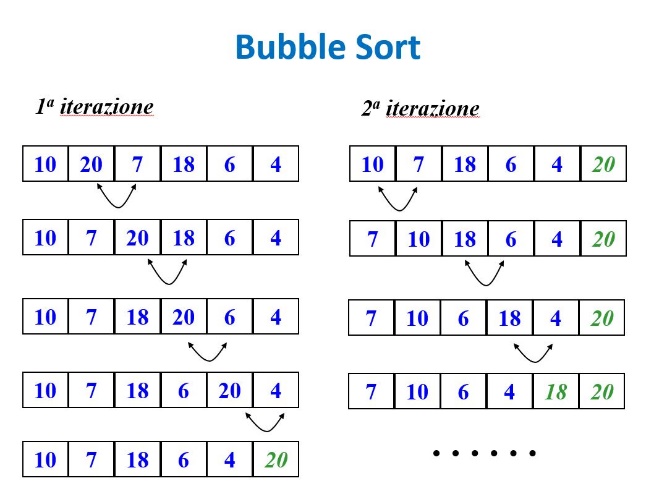
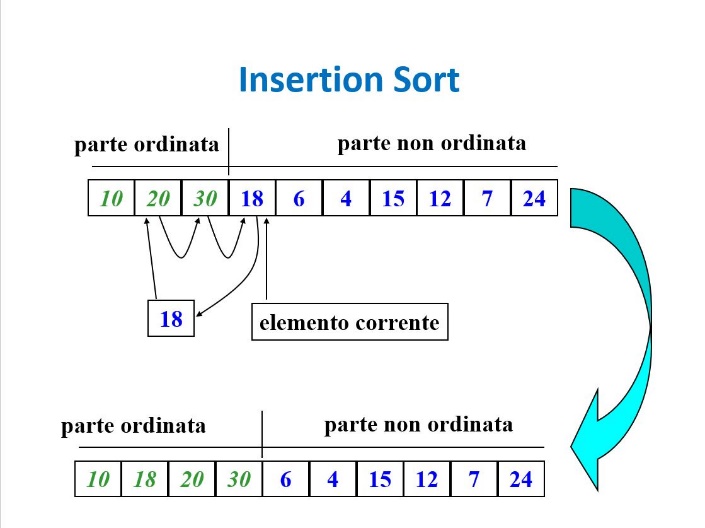
Scrivere un programma che chiede all’utente di inserire il proprio nome e cognome e che allochi dinamicamente **lo spazio necessario** a memorizzare i due array di caratteri separatamente. Successivamente si pongano le due stringhe in una matrice.

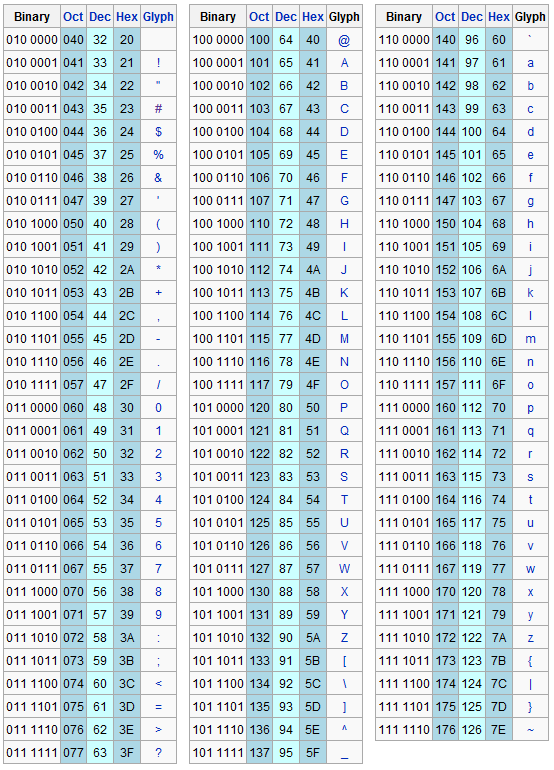
A questo punto implementare le seguenti:

1. una funzione char\* merge(char\*, char\*) che prende in input il nome e il cognome e produca una nuova stringa che è il risultato della concatenazione delle due date in input  
   *[Alternativa]*: *char\* merge(char\*\*)*
2. una funzione char\* incastona(char\*, char\*) che produce una nuova stringa secondo la seguente regola: la più piccola delle due va *incastonata* nella più lunga sostituendo i caratteri di questa ultima. Esempio: a=”fabio” b=”narducci” restituirà in output la stringa c= “nfabioci” (*nota. quando la stringa più lunga ha un numero pari di caratteri, la stringa più corta viene incastonata al centro a meno di un carattere a sx o dx*).  
   *[Alternativa]*: *char\** incastona*(char\*\*)*
3. una funzione void inverti(char\*\*) che prende la matrice e rimpiazza la stringa del nome con quella del cognome e viceversa (in sostanza uno swap delle due stringhe)
4. una funzione void crea (char\*\*) che aggiunge una terza stringa alla matrice composta dai primi tre caratteri del nome e gli ultimi tre del cognome. Fare uso delle funzioni di copia di stringhe e, possibilmente, aritmetica dei puntatori.
5. una funzione void ordina(char \*) che ordini i caratteri del nome e del cognome con un algoritmo di ordinamento a scelta tra *InsertionSort* o *BubbleSort*.

Alcune considerazioni:

* Dal momento che non è nota a priori la dimensione dell'array, è opportuno gestire l'allocazione dell'array in maniera dinamica.
* Convertire e gestire opportunamente i due diversi tipi di dato **char** e **int** e relativi puntatori

****

****