Analisi:

# lezione 1: Liste concatenate

## lavoro:

Creare un array di Struct contenente ognuna la posizione del valore successivo e un contenuto come un nome sotto forma di stringa.  
In questo modo si ottiene una lista.  
I primi 10 valori sono assegnati da codice, gli altri 10 per esempio sono spazi vuoti pronti per essere riempiti.  
Inizializziamo 10 valori, che per semplicità usiamo 10 compagni di classe a caso.

## Opzioni menù:

* Inserimento.
* Rimozione.
* Visualizzazione Fisica.
* Visualizzazione Logica.
* Ricerca Valore.

## Struttura struct:

Chiamo la Struct “dato”.  
Chiamo il vettore di Struct “dati”.

La struttura delle struct sarà la seguente:

* Contenuto.
* Pos\_Suc.

Lo scopo del contenuto è quello di immagazzinare un valore sotto forma di stringa, in questo caso il nome o cognome del compagno di classe.

Lo scopo della posizione successiva è quello di puntare alla posizione in cui si trova il nome successivo nell’alfabeto.

## Inizializzazione:

Inserisco tramite codice i valori nel vettore di Struct, con le posizioni prefissate.

## Inserimento:

Controllo se ci sono spazi vuoti, in caso positivo continuo.  
Chiedo all’utente il valore da inserire.  
Successivamente confronto il valore da inserire e capisco in ordine alfabetico la sua posizione in cui si dovrebbe trovare, lo inserisco nel primo spazio vuoto con valore “null” e posizione puntata -1, e modifico il valore a lui precedente (in ordine alfabetico) in modo che lo punti alla sua posizione, e poi faccio in modo che lui stesso punti al suo successivo, nel caso non ci fosse, punterà al primo spazio vuoto che trova.  
Nel caso particolare in cui il suo valore è il più piccolo mai inserito, la variabile globale logica per la lettura del primo valore deve essere modificata con la posizione del valore appena inserito.

## Rimozione:

Verifico subito una condizione di criticità, ossia se il nome da rimuovere inserito è il primo valore della lista, in caso affermativo ottengo la sua posizione puntata e la setto come nuovo valore minimo globale da cui iniziare la lettura logica (la posizione del primo elemento in sintesi) e poi setto ai valori di default “nulli” il valore alla posizione da rimuovere.

In caso normale invece, in cui l’elemento da rimuovere non sia il primo, tramite un ciclo while che continua fino a quando viene il valore viene trovato oppure la posizione puntata è quella nulla di riferimento (attualmente -1), cerco il valore leggendo la posizione di quello attuale in lettura logica e cercando con questa il valore del successivo, se corrisponde a quello da cercare, eseguo le azioni di modifica del puntatore del valore del ciclo attuale con quello del puntatore del ciclo “successivo” letto prima, e setto ai valori nulli di default il valore da rimuovere.

## visualizzazione fisica:

Devo visualizzare i valori in base all’ordine fisico e non a quello logico a cui puntano, quindi tramite un ciclo for per il numero di elementi visualizzo il contenuto, oppure uso un ciclo while che continua fino a quando non viene visualizzato un valore bandiera “null” e posizione 0 che significa che è vuoto.

## Visualizzazione logica:

Devo visualizzare i valori in base alla posizione a cui ognuno punta, partendo dal primo fino all’ultimo che punterà ad una posizione contenente un valore nullo che quindi non sarà visualizzato, questo posso farlo tramite un ciclo while che continua fino a quando un valore “null” e posizione 0 non venga trovato.

## Ricerca valore:

Chiedo all’utente di inserire il nome dell’utente e tramite un ciclo usando le posizioni a cui ogni elemento punta, cerco se la stringa del nome o cognome è contenuta, in caso positivo procedo comunico all’utente in che posizione logica è stato trovato, nel caso non venga trovato comunico all’utente che il valore inserito non è contenuto nel vettore di Struct.