**Linguaggio di Programmazione, Interpretazione e Libreria**

La libreria dipende da chi “fornisce” il sistema, quindi dipende da sistema a sistema, per questo motivo a lezione abbiamo visto dei risultati diversi con lo stesso programma, eseguito prima sotto Linux e dopo sotto Windows. In Windows, la funzione printf è fuori dallo standard e quindi non riconosce alcuni caratteri come l’hh.

**#define \_\_USE\_MINGW\_ANSI\_STDIO 1** mi permette di andare ad utilizzare la libreria Standard piuttosto che quella del sistema in cui operiamo, in questo caso quella di Windows.

**Visibilità variabili e persistenza in memoria**

**Variabile automatica**: variabile che viene allocata all’inizio di un blocco e viene rimossa alla fine. A livello di visibilità è visibile nel blocco ed in tutti i sotto blocchi contenuto nel blocco principale. Però non vale il contrario, cioè se dichiaro una variabile in un sotto blocco, non è accessibile dai blocchi presenti a livelli superiori.

**Variabile globale**: viene allocata all’inizio dell’esecuzione del programma e viene rimossa al termine. E’ visibile in tutti i blocchi e sotto blocchi di codice. Per rendere una variabile globale, basta dichiararla all’esterno di una qualsiasi funziona.

**Variabile globale static**: static, la

memoria persiste durante tutto quanto il programma, però la visibilità è ristretta all’unità di compilazione, cioè è visibile solamente dal file sorgente dove è presente e non da altri file tramite la compilazione incrociata.

**extern** -> Quando compilo due file assieme, e devo prendere una variabile dall’altro file.

**Puntatori C**

Passaggio per riferimento nelle funzioni, non supportato nativamente ma possiamo utilizzare i puntatori

Un puntatore rappresenta un indirizzo di memoria