

Lezione 4

Le stringhe sono semplicemente array monodimensionali di caratteri, il placeholder da utilizzare è %s.

La stringa non è una variabile semplice quindi nel C bisogna utilizzare la libreria <string.h>.

Se si inserisce una stringa superiore al numero dichiarato la stringa inserita verrà troncata, esiste anche la libreria <ctype.h> che è molto utile per quanto riguarda la gestione delle stringhe, come per esempio controllare se una stringa contiene una lettera in una determinata posizione o un numero.

Le funzioni e le procedure sono dei “pezzi” di codice che possono essere utilizzate più volte.

Le procedure sono dei sottoprogrammi il cui compito è quello di produrre un effetto (può anche produrre un risultato), mentre una funzione è un sottoprogramma che ha il compito di produrre un risultato (avrà istruzione “return”).

In C la creazione delle funzioni e delle procedure avviene utilizzando lo stesso costrutto.

Procedura → void procedura(int a, int b).

Funzione → int max(int a, int b);

In C la dichiarazione di una funzione o di una procedura prende il nome di un prototipo, essa è obbligatoria e deve essere inserita subito dopo le direttive del precompilatore e prima del main.

I parametri sono utilizzati per costruire sottoprogrammi più versatili che possono essere utilizzati in contesti completamente differenti, gli argomenti possono essere:

- Input → Passare informazioni ad un sottoprogramma;
- Output → Ricevere informazioni da un sottoprogramma.

```
/*Se non vengono passati degli argomenti è preferibile usare void per indicare che è fatto apposta*/
/* Usare prototipi */
```

Naturalmente bisogna verificare le corrispondenze tra parametri formali e parametri attuali relativamente a:

N → Numero di argomenti;

O → Ordine degli argomenti;

T → Tipo di ciascun argomenti;

La funzione rand() si trova nella libreria stdlib.h e genera un valore casuale, la funzione rand() non ha un seme, per renderla pseudo-casuale bisogna utilizzare la funzione srand() che genera il seme, all'interno della srand bisogna passare il parametro time(NULL) (presente nella libreria time.h) che fa un riferimento all'orario attuale e quindi genera un numero casuale su cui srand genererà un seme e verrà generato il numero pseudo-casuale.

```
/* Non esiste la vera casualità in informatica, infatti viene sempre applicato un algoritmo */
```

Le variabili di tipo primitivo int, float, char ecc sono passati per valore mentre per quanto riguarda le variabili complesse invece sono passate per valore.

Quando una procedura deve restituire un risultato è necessario che si usi il passaggio per referenza (o indirizzo).

L'operatore * dichiara che il parametro formale indicato è un puntatore ad una variabile del tipo indicato prima del suo nome, la dichiarazione di un puntatore comunica al compilatore che la variabile dichiarata conterrà l'indirizzo della variabile del tipo indicato.

Il puntatore si utilizza attraverso l'utilizzo della & commerciale che passa l'indirizzo, se si utilizza l'operatore * prima del puntatore indica il valore che è puntato dal puntatore.

Questi si utilizzano per restituire più output dal sottoprogramma.

La istruzione scanf() inoltre restituirà quanti elementi sono ricevuti in input quindi può essere utilizzata per controllare il numero degli input.

```
/*  
Controllo = scanf("%d %d %d", &primo, &secondo, &terzo);  
if (controllo != 3)  
{  
    printf("Corretto");  
}  
else  
{  
    printf("Errato");  
}
```