

### Tipi di dati

<b>Int</b>	interi
<b>Float</b>	virgola mobile
<b>Double</b>	virgola mobile a precisione doppia
<b>Char</b>	singolo carattere
<b>Void</b>	indefinito

### Errori di sintassi

Errori che si verificano durante la creazione di un programma.

### Incremento e decremento

Incremento	++
Decremento	-

### Parole chiave

Parole che fanno parte della libreria del linguaggio C standard

**Esempio:** `void, return int`

### Dangling Pointer

Puntatore *penzolante*: puntatore che punta a un indirizzo di memoria di qualsiasi variabile, ma dopo un po' la variabile è stata cancellata dalla posizione di memoria mantenendo il puntatore che punta a quella posizione

### Funzione static

Mi permette di creare variabili persistenti in una funzione. Le variabili interne delle funzioni sono variabili locali. Una volta terminata l'esecuzione della funzione, il loro valore si perde. L'istruzione static mantiene in memoria il valore di una variabile interna della funzione dopo ogni chiamata

### Funzioni abs() e fabs()

<code>abs()</code>	valore assoluto di interi
<code>fabs()</code>	valore assoluto di floating point

### Wild Pointer

Puntatori non inizializzati

### Puntatore

Variabile che contiene l'indirizzo di memoria di un'altra variabile

Per definire un puntatore è necessario anteporre al nome della variabile un asterisco, che viene chiamato operatore di indirezione o deferenza

**Esempio:** `int a = 5, *x=&a, **y=&x`  
`;` Qui `**y` restituisce il valore della variabile `a`.

### ++a e a++

<code>++a</code>	incremento avviene per primo su una variabile
<code>a++</code>	incremento avviene dopo il valore di una variabile

### Caratteristiche C

<b>Portabilità</b>	è un linguaggio indipendente dalla piattaforma
<b>Modularità</b>	possibilità di suddividere grandi programmi in piccoli moduli
<b>Flessibilità</b>	possibilità di controllare il linguaggio
<b>Velocità</b>	C viene fornito con il supporto per la programmazione di sistema
<b>Estensibilità</b>	possibilità di aggiungere nuove funzionalità

### Differenza tra virgolette " e parentesi <>

`#include "path"` il preprocessore cerca i file di inclusione nel seguente ordine:  
 1. nella directory del file  
 2. nelle directory "-padre" del file  
 3. lungo il percorso specificato da ogni opzione del compilatore  
 4. lungo i percorsi specificati dalla `include`

`#include <path>` il preprocessore cerca i file di inclusione nel seguente ordine:  
 1. lungo il percorso specificato da ogni opzione del compilatore  
 2. lungo i percorsi specificati dalla `include`

### Commenti

<b>Spiegazione</b>	isolano una parte del codice affinché non influisca durante il debug
<code>//</code>	commento su una sola riga
<code>/*-----*/</code>	commento su più righe

### Istruzione Break

Termina l'esecuzione dell'istruzione contenuto in un loop o nello switch

### Funzione in C

<b>Definizione</b>	contiene 4 sezioni principali
<b>Tipo di ritorno</b>	tipo di dati del valore di ritorno della funzione



### Funzione in C (cont)

<b>Nome funzione</b>	nome che si sceglie
<b>Parametri</b>	valori di input per la funzione
<b>Body</b>	il corpo, la raccolta di istruzioni che deve eseguire la funzione

### Ciclo annidato

Ciclo che viene eseguito all'interno di un altro ciclo. Il primo loop è chiamato **outer loop** e il loop interno è detto **inner loop**. Quest'ultimo viene eseguito il numero di volte definito in un ciclo esterno.

### Tipi di ciclo

<b>Cosa sono</b>	istruzioni che consentono l'esecuzione di istruzioni o gruppi di istruzioni in modo ripetuto
<code>for</code>	crea un ciclo iterativo
<code>while</code>	mentre avviene l'istruzione continua il programma
<code>do-while</code>	viene eseguito il corpo dell'istruzione e poi viene valutata
<i>ciclo annidato</i>	ciclo in un altro ciclo

### File di intestazione

File contenente le definizioni e i prototipi delle funzioni utilizzate. Noto anche come **file di libreria**.

Contiene comandi come `printf` e `scanf` e proviene dal file di libreria `stdio.h`.

### Natura ciclica tipi di dati

Se viene assegnato un valore oltre l'intervallo del tipo di dati, non ci sarà alcun errore del compilatore e il valore cambia secondo un ordine ciclico. Questo vale per i tipi di dati: **char**, **int**, **long**.

Non vale per **float**, **double**, **long double**

### Funzione prototipo

Dichiarazione di una funzione con le seguenti informazioni: nome funzione, tipo restituito della funzione, elenco parametri della funzione.

**Esempio:** `int Sum(int, int);`

### Simboli = e ==

<code>==</code>	operatore di confronto
<code>=</code>	operatore di assegnazione

### Programmazione modulare

Basata su più files di codice sorgente, ognuno dei quali dedicato alla risoluzione di un determinato problema.

### Int

Il tipo di dati **int** è in grado di memorizzare solo valori compresi tra -32768 e 32767.

### Sequenza di escape

<code>\n</code>	nuova riga
<code>\b</code>	backspace
<code>\t</code>	tabulazione orizzontale

### Comando rand()

Serve a generare numeri casuali.

### File ad accesso sequenziale

I dati vengono salvati in uno schema sequenziale. Quando si recuperano i dati da tali file, ogni dato viene letto uno alla volta finché non vengono trovate le informazioni richieste.

### Stack

Struttura dati lineare che funge da raccolta di elementi con 3 operazioni principali:

1. **Push** operazione, che aggiunge un elemento allo stack
2. **Pop** operazione, che rimuove l'elemento aggiunto più di recente che non è stato ancora rimosso
3. **Peek** operazione, che restituisce l'elemento superiore senza modificare lo stack.

Il **push** e il **pop** avvengono solo ad un'estremità della struttura, denominata **top** della stack. L'ordine in cui gli elementi escono da una stack dà origine al suo nome alternativo, **LIFO** (Last-In, First-Out).

### Algoritmo

Successione di istruzioni o passi che definiscono le operazioni da eseguire sui dati per ottenere i risultati.

### Funzione Toupper()

Viene utilizzata per convertire il valore in maiuscolo quando viene utilizzato con i caratteri.

### Modificatore

Un prefisso al tipo di dati di base che viene utilizzato per indicare la modifica per l'allocazione dello spazio di archiviazione.

**Esempio:** In un processore a 32 bit, lo spazio di archiviazione per il tipo di dati `int` è 4. Quando usiamo il modificatore cambia lo spazio come segue:

**Long int:** lo spazio di archiviazione è di 8 bit

**Short int:** lo spazio di archiviazione è di 2 bit

